ปีที่ 2 ฉบับที่ 9 กรกฎาคม 2527

ไมโคร อิเลคทรอนิคส์

วารสารไมโครคอมพิวเตอร์ฉบับแรกในประเทศไทย



ฉบับปรับปรุงใหม่

IBM PC jr ดาวรุ่งดวงใหม่ หรือ ลูกเป็ดขี้เหร่

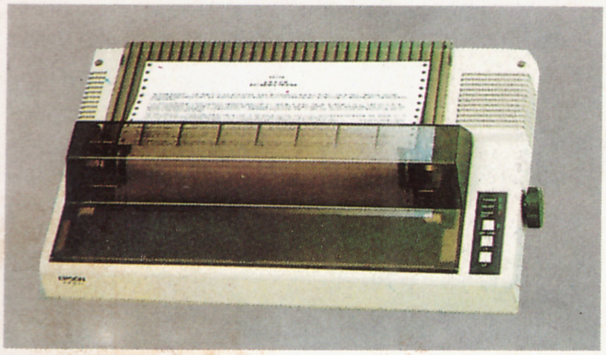
- Virtual Memory
- MC 68000
- Window Software
- MS-DOS : 2.0

฿ 7.00

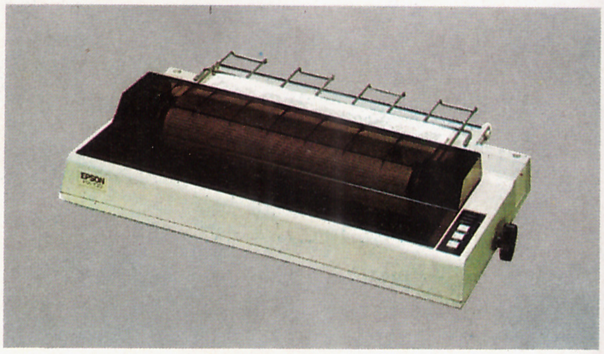
20.

EPSON: Last Word in Technology

SIC



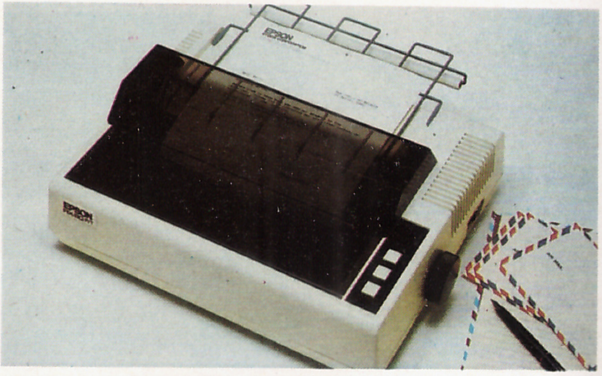
Dynamic Printer FX-100
160 cps printing speed, 136 columns per line (normal mode), Tractor/friction paper feed



Upgraded Printer RX-100
100 cps printing speed, 136 columns per line (normal mode), Tractor/Friction feed.



Versatile Printer FX-80
160 cps printing speed. 80 columns per line (normal mode). Tractor/friction paper feed (cut sheet or roll paper)



Personal Printer with Friction Feed RX-80 F/T
100 cps printing speed. 80 columns per line (normal mode). Tractor/friction paper feed (cut sheet or roll paper)



AP-200 Ultraslim Floppy Disk Unit
100% Compatible with Apple II and Apple II+, Single-side 5¼" Floppy Disk Drive, 143 KB (formatted) per drive or 286 KB total



Portable Computer HX-20
Ram 16 KB (expandable to 32 KB)
Rom 32 KB (expandable to 72 KB)
Built-in Peripheral: Display, Printer, clock, Music Generator, RS 232 C Serial Interface

SIC

บริษัท สหวิริยาอินเตอร์แนชชั่นแนลคอมพิวเตอร์ จำกัด
28/1 ตึกประภาวิทย์ ชั้น 3 ถนนสุรศักดิ์ สีลม กรุงเทพมหานคร
โทร. 235-1887, 233-9878-9

ไมโคร อิเลคทรอนิคส์

ปีที่ 2 ฉบับที่ 9
กรกฎาคม 2527

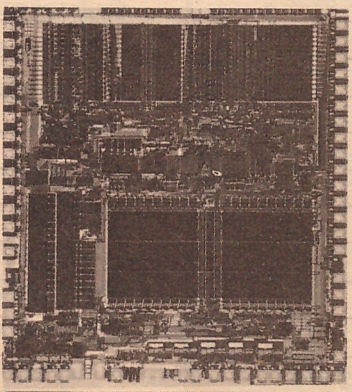
วารสารไมโครคอมพิวเตอร์ฉบับแรกในประเทศไทย



ไมโคร อิเลคทรอนิคส์

ฉบับที่ ๑ กรกฎาคม

บทความน่าอ่าน



Hardware

คอลัมน์ประจำ

ไมโคร
อิเลคทรอนิคส์

เจ้าของ
บริษัท ไมโครมีเดีย จำกัด
สำนักงาน
1091/230 ศูนย์การค้านครหลวง เพชรบุรีตัดใหม่
กรุงเทพฯ 10400 โทร. 2522070

บรรณาธิการบริหาร
เสริมพร พูลทรัพย์
บรรณาธิการจัดการ
วศิน เพิ่มทรัพย์
สุวัฒน์ ปุณณชัย
ที่ปรึกษา
ยรรยง เต็งอำนวย, ไตรรัตน์ ใจสำราญ
ปริทรรศน์ พันธุบรรยงก์
กองบรรณาธิการ
ประพันธ์ ธีระวรรณวิไล ดนัยวัฒน์ กิตตินันท์
นพดล ปัญญาธิปัตย์ สมคิด บุญญประภากร
สิริชัย ฟูตระกูล อดิสร กรุงเกษม
บรรณาธิการฝ่ายศิลป์
จารึก อรรถกรศิริโพธิ์
ฝ่ายศิลป์
ปรียา จันเฉียวฉุน
ฝ่ายธุรการ
สุปราณี เตชปัญญาสาร สุมาลี ธรรมเจริญ
ผู้จัดการฝ่ายโฆษณา
เสริมพร พูลทรัพย์
ฝ่ายโฆษณา
วาสนา ช่วยแก้ว
แยกสี
สยามอาร์ต
จัดจำหน่าย
เพ็ญบุญ โทร. 278-1317

วัตถุประสงค์

1. นำเสนอวิทยาการคอมพิวเตอร์ทุกด้าน ให้แก่ผู้อ่านทุกระดับ
2. เป็นสื่อกลางในการเผยแพร่ข่าวสารข้อมูลในวงการคอมพิวเตอร์

สวัสดีครับคุณผู้อ่าน สำหรับในฉบับนี้ผมมารับหน้าที่แทนคุณเสริมพรเป็นการชั่วคราว พร้อมกับการเปลี่ยนแปลงหลาย ๆ อย่าง ซึ่งคุณผู้อ่านก็คงจะสังเกตเห็นบ้างแล้ว

กว่าปีมาแล้ว นับตั้งแต่เวลาที่คุณเสริมพรกับผมและผู้ร่วมงานคนอื่น ๆ ช่วยกันเข็นไมโครฯเล่มแรกออกสู่สายตาผู้อ่าน ในขณะนั้นตลาดวารสารคอมพิวเตอร์ยังว่างเปล่า นอกจากคอมพิวเตอร์สารของสมาคมคอมพิวเตอร์แห่งประเทศไทยแล้วก็มีเพียงเราเท่านั้น ในขณะนั้นอีกเช่นกันที่ความสนใจของผู้คนทั่วไปเริ่มหันเข้าหาคอมพิวเตอร์ วิทยาการคอมพิวเตอร์ในบางรูปแบบเริ่มเผยแพร่สู่ประชาชนโดยเฉพาะเทคโนโลยีทางด้านอิเลคทรอนิคส์อันเป็นพื้นฐานของดิจิตอลคอมพิวเตอร์ก็ได้เริ่มปรากฏเป็นบทความในวารสารอิเลคทรอนิคส์ต่าง ๆ ส่วนทางด้านซอฟท์แวร์หนังสือคู่มือการเขียนโปรแกรมภาษาต่าง ๆ ก็ได้มีการจัดพิมพ์ขึ้นมากมายหลายเล่มทั้ง FORTRAN, COBOL, BASIC และอื่น ๆ อีกมาก แต่ก็ยังไม่มีใครจัดทำวารสารที่เกี่ยวกับคอมพิวเตอร์โดยเฉพาะที่เน้นการนำวิทยาการด้านนี้มาสนองความอยากรู้ของประชาชนทั่วไปออกมา

นี่เองคือจุดกำเนิดของไมโครฯ เราเป็นกลุ่มคนทำหนังสือเล็ก ๆ น้อยทั้งผู้คน การสนับสนุนและทุนรอน แต่มีความตั้งใจที่ไม่เล็ก ในอันที่จะนำวิทยาการคอมพิวเตอร์ อันเป็นเทคโนโลยีที่ก้าวหน้าไปอย่างไม่หยุดยั้งและมีบทบาทมากขึ้นเรื่อยในชีวิตคนปัจจุบันมาเสนอต่อคุณผู้อ่าน เราทำหนังสือก็เพราะหวังว่าจะมีคนอ่านสิ่งที่เราทำ และได้ประโยชน์จากมันบ้างพอสมน้ำสมเนื้อกับเงินที่ต้องจ่ายไป

ถ้าจะถามว่าทำไมถึงต้องเป็น "ไมโครอิเลคทรอนิคส์" ? เรื่องนี้เป็นเพราะเราต้องการจะทำหนังสือที่จะเป็นตัวเชื่อมระหว่างเทคโนโลยีทางฮาร์ดแวร์ ในวารสารอิเลคทรอนิคส์ในระยะนั้น กับส่วนที่ยังว่างอยู่มากของเรื่องราวทางซอฟท์แวร์และการประยุกต์ใช้งาน (application) ตลอดจนทางด้านข่าวคราว แนวความคิดและผลิตภัณฑ์ใหม่ ๆ ในวงการอุตสาหกรรมคอมพิวเตอร์ นั่นคือที่มาของคำว่า "อิเลคทรอนิคส์" ในชื่อหนังสือ รวมกับ "ไมโคร" อันหมายถึงไมโครคอมพิวเตอร์ซึ่งเป็นผู้ก่อให้เกิดการ Boom หรือการขยายตัวอย่างขนานใหญ่ของคอมพิวเตอร์ในปัจจุบัน และในอีกความหมายหนึ่ง "ไมโครอิเลคทรอนิคส์" หมายถึงเทคโนโลยีในการผลิตอุปกรณ์อิเลคทรอนิคส์ขนาดจิ๋วที่เป็นส่วนประกอบสำคัญของคอมพิวเตอร์ ถ้า CPU เป็นหัวใจของระบบคอมพิวเตอร์ ไมโครอิเลคทรอนิคส์ก็คือแต่ละเซลที่ประกอบกันขึ้นเป็นหัวใจนั้น และก็เป็นเช่นเดียวกับความตั้งใจของเราที่จะนำเสนอเนื้อหาสาระที่เป็นหัวใจหรือแก่นสารของวิทยาการคอมพิวเตอร์สู่คุณผู้อ่านด้วย





เราจะไปทางไหนกัน ?

ในเวลากว่าปีที่ผ่านมา เราพยายามอยู่เสมอที่จะบรรลุวัตถุประสงค์อันนั้น แม้ว่าจะมีอุปสรรค ความผิดพลาดและความผิดหวัง ตลอดจนความล่าช้าอันเนื่องจากสาเหตุนานัปการ แต่เราก็ไม่ย่อท้อที่จะยืนหยัดต่อไป การสนับสนุนของคุณผู้อ่านเป็นกำลังใจที่สำคัญของเรา แม้จะถูกติเตียน บ่นว่า ว่าทำไมหนังสือออกช้า ทำไมพิมพ์ผิด ๆ ถูก ๆ ตก ๆ หาย ๆ เขียนอะไรอ่านไม่รู้เรื่อง เขียนอะไรไม่มีสาระ ฯลฯ เราก็ขอน้อมรับและพยายามแก้ตัว เพราะเรารู้ว่าเมื่อมีผู้ติเตียน หมายถึงว่าเขา ผิดหวังแต่การผิดหวังก็ย่อมแสดงว่าเขาตั้งความหวังไว้ที่เรา ดังนั้นเราจะพยายามแก้ตัวใหม่ไม่ให้เขาผิดหวังอีก

หน้าที่อีกประการหนึ่งของวารสารคอมพิวเตอร์ ไม่เพียงแต่นำเสนอวิทยาการเท่านั้น ยังต้องเป็นสื่อกลางของวงการด้วยระหว่างผู้ซื้อ ผู้ใช้ ผู้ผลิต ผู้ขายและบุคคลอื่น ๆ อีกมากมาย ทั้งข่าวคราว ผลิตภัณฑ์ การใช้งาน และอื่น ๆ อีกหลายหลากที่ผู้คนในวงการนี้รวมทั้งผู้คนทั่วไปสนใจ เราก็ได้พยายามที่จะทำหน้าที่ทางด้านนี้ให้สมกับความไว้วางใจที่ทุกฝ่ายมอบให้เราลงโฆษณาให้แก่ผู้ผลิต ผู้ขาย ขณะเดียวกันเราก็พยายามเสนอบทความที่ให้ข้อเท็จจริงแก่ผู้ซื้อและให้ข้อมูลแก่ผู้สนใจทั่วไป หลายครั้งที่เราส่งนักเขียนไปทดสอบเครื่องคอมพิวเตอร์ตามบริษัท เช่น EPSON, ATARI หลายครั้งที่เราได้ข้อมูลจากผู้ร่วมงานของเราที่เผชิญทำงานอยู่กับผู้ขาย นั่นเป็นความมั่นใจของเราที่ว่าเราได้ข้อเท็จจริง ๆ ล้วนมาเสนอคุณผู้อ่าน และหลายครั้งที่เราจำเป็นต้องอาศัยข้อมูลจากผู้ขายโดยตรงกับข้อมูลจากวารสารต่างประเทศมาประมวล กลั่นกรอง แยกแยะข้อเท็จจริง เพื่อว่าถ้าข้อมูลเหล่านั้นผิดก็จะผิดเพียงน้อยที่สุดเท่าที่จะเป็นไปได้ และเป็นเหตุสุดวิสัยที่เกินจากความพยายามของเราแล้วเท่านั้น

กว่าปีที่ผ่านมาได้สร้างสมประสบการณ์หลาย ๆ อย่างให้กับเรา ซึ่งเราได้นำมันมาพิจารณาทบทวนว่า สิ่งใดบ้างที่ควรจะทำเพื่อให้ผลงานของเราดีขึ้นและเราก็ได้ตัดสินใจที่จะทำการปรับปรุงครั้งใหญ่ดังที่คุณผู้อ่านเห็นอยู่นี้ สิ่งที่เราพยายามปรับปรุงก็คือเนื้อหาสาระของบทความ หลายเรื่องที่เราได้ตัดออกไป เช่นคู่มือการใช้งาน SuperCalc, dBASE II ทั้งนี้เนื่องจากเราไม่อาจจัดเสนอบทความดังกล่าวในรูปแบบที่ทันต่อเหตุการณ์หรือยุคสมัยของ Application Software ที่เปลี่ยนแปลงไปอย่างรวดเร็วได้ นอกจากจะนำเสนอในรูปแบบของเทคนิคการใช้งานหรือวิธีการดัดแปลง/ประยุกต์ใช้งานแบบใหม่ที่น่าสนใจ ซึ่งเราจะพยายามจัดทำในโอกาสต่อ ๆ ไป บทความบางเรื่องที่จัดอยู่ในลักษณะเดียวกันนี้เช่น WordStar หรือ Applesoft Compiler ก็จะจบลงในฉบับนี้และฉบับหน้าตามลำดับ ส่วนบทความที่เราจะเน้นหนักในการนำเสนอในวารสารไมโครต่อ ๆ ไปจะอยู่ในลักษณะของสาระวิทยาการที่น่าสนใจกว่านั้น โดยจะพยายามเจาะให้ลึกและครอบคลุมให้กว้าง มากกว่าการจะเน้นรายละเอียดวิธีการใช้งาน Application software แต่ละตัว ซึ่งผู้อ่านหรือผู้ใช้จะสามารถหาได้จากแหล่งอื่นเช่นหนังสือคู่มือการใช้งานที่เป็นเล่ม ๆ หรือผู้ขายเองอยู่แล้ว

นอกจากนี้อีกสิ่งหนึ่งที่เราพยายามปรับปรุงก็คือทางด้านการจัดรูปเล่มซึ่งทางฝ่ายศิลป์ได้เพิ่มความพิถีพิถันในการจัดทำให้มากขึ้น เพื่อลดความบกพร่องไม่งามหรือดูหนักหนาเกินไป ตามที่คุณผู้อ่านบ่นมา อย่างไรก็ตามก็ยังอาจมีข้อบกพร่องใด ๆ หลงหูหลงตาไปบ้างก็ต้องขออภัยไว้ด้วย และหากท่านผู้อ่านมีความคิดหรือเห็นข้อติชมอย่างไรก็ขอให้เราได้มีส่วนรับรู้ด้วยจะเป็นพระคุณอย่างยิ่ง เพื่อที่ว่าเราจะได้ปรับปรุงให้ผลงานของเราดียิ่งขึ้นในโอกาสต่อ ๆ ไป

นอกจากความคิดเห็นและข้อติชมแล้ว ท่านผู้อ่านยังสามารถที่ส่วนร่วมกับเราได้อีกหลายทาง เรายินดีเสมอที่จะรับพิจารณาบทความ โปรแกรม หรือแม้แต่ข้อมูลใด ๆ ที่น่าสนใจ และอาจเป็นประโยชน์ต่อผู้อ่านอื่น ๆ โดยมีค่าตอบแทนให้ตามสมควร หรือถ้าคุณมีแต่เพียงแนวความคิดหรือข้อมูล แต่ไม่รู้จะเขียนออกมาเป็นบทความอย่างไร ก็ลองมาคุยกับเราดูซิครับ เรายินดีจะแลกเปลี่ยนความคิดเห็นและให้คำแนะนำแก่คุณเท่าที่จะทำได้



ส่วนคุณที่มีปัญหาคอลัมน์ "สารพันปัญหา คอมพิวเตอร์" ก็ไม่ได้หายไปไหน จะกลับมาพบกับคุณผู้อ่านตามปกติในฉบับต่อ ๆ ไป แต่สำหรับเรื่องของปัญหานี้ ผมใคร่ขอทำความเข้าใจกับคุณผู้อ่านสักหน่อยว่า เราจะพยายามตอบคำถามอย่างกระจ่างชัดและละเอียดเท่าที่จะทำได้ แต่เนื่องจากเนื้อที่จำกัด จึงจำเป็นต้องเลือกตอบเฉพาะคำถามที่น่าสนใจ หรืออาจเป็นประโยชน์ต่อคุณผู้อ่านท่านอื่น ๆ ด้วย นอกจากนี้บางปัญหาก็ตอบได้ยาก-ง่ายต่างกันมา ดังนั้นบางคำถามเราจึงสามารถตอบได้อย่างรวดเร็ว แต่บางคำถามก็ต้องขอเวลาค้นคว้าและสอบถามจากผู้รู้บ้างเพื่อให้ได้คำตอบที่ถูกต้องจริง ๆ

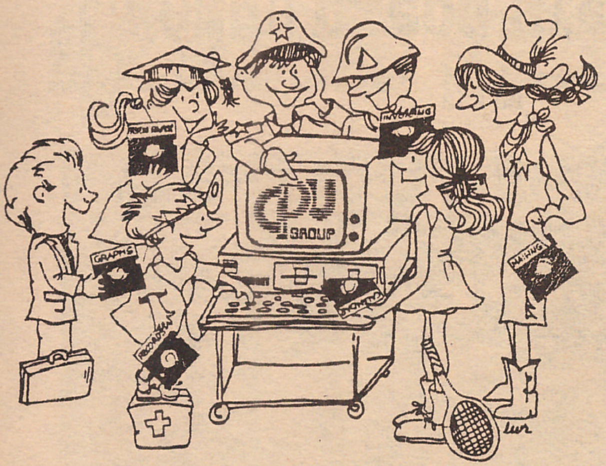
เท่าที่เล่ามาก็หวังว่าจะเป็นการตอบคำถามที่จั่วหัวเรื่องไว้ว่า "เราจะไปทางไหนกัน?" ได้พอสมควรนะครับ เรากำลังปรับปรุงอยู่เรื่อย ๆ เพื่อให้บรรลุถึงเป้าหมายที่วางไว้ คือการนำเสนอวิทยาการคอมพิวเตอร์ในทุก ๆ ด้านแก่คุณผู้อ่าน และเป็นสื่อกลางของวงการ และตราบเท่าที่คุณผู้อ่านยังให้ความสนับสนุนเราจะพยายามทำหน้าที่ของเราให้ดีที่สุด



พบกันใหม่ฉบับหน้าสวัสดีครับ

วศิน เพิ่มทรัพย์

and User GROUP



เปิดแล้ว...!

เรา Computer Programmer and User GROUP
เปิดรับสมาชิกผู้ใช้และผู้ที่สนใจไมโครคอมพิวเตอร์

ท่านเป็นสมาชิก... จะได้บริการ

- COPY PROGRAM ฟรี
- การอบรมทางวิชาการ
- เป็นสื่อกลางระหว่ งสมาชิก
- ส่วนลดพิเศษในกิจกรรมต่าง ๆ
- ใช้เครื่องฯ ในราคาพิเศษ
- ข่าวสารและวิชาการจากเราและสมาชิกเป็นประจำ
- อื่น ๆ ...

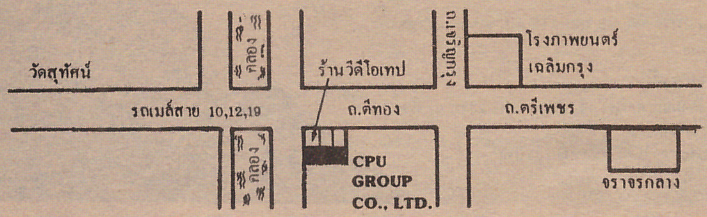
RX-8800

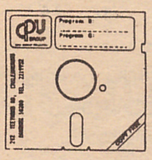


PC-5500



เปิดบริการทุกวัน
ไม่เว้นวันหยุด
9.00-21.00



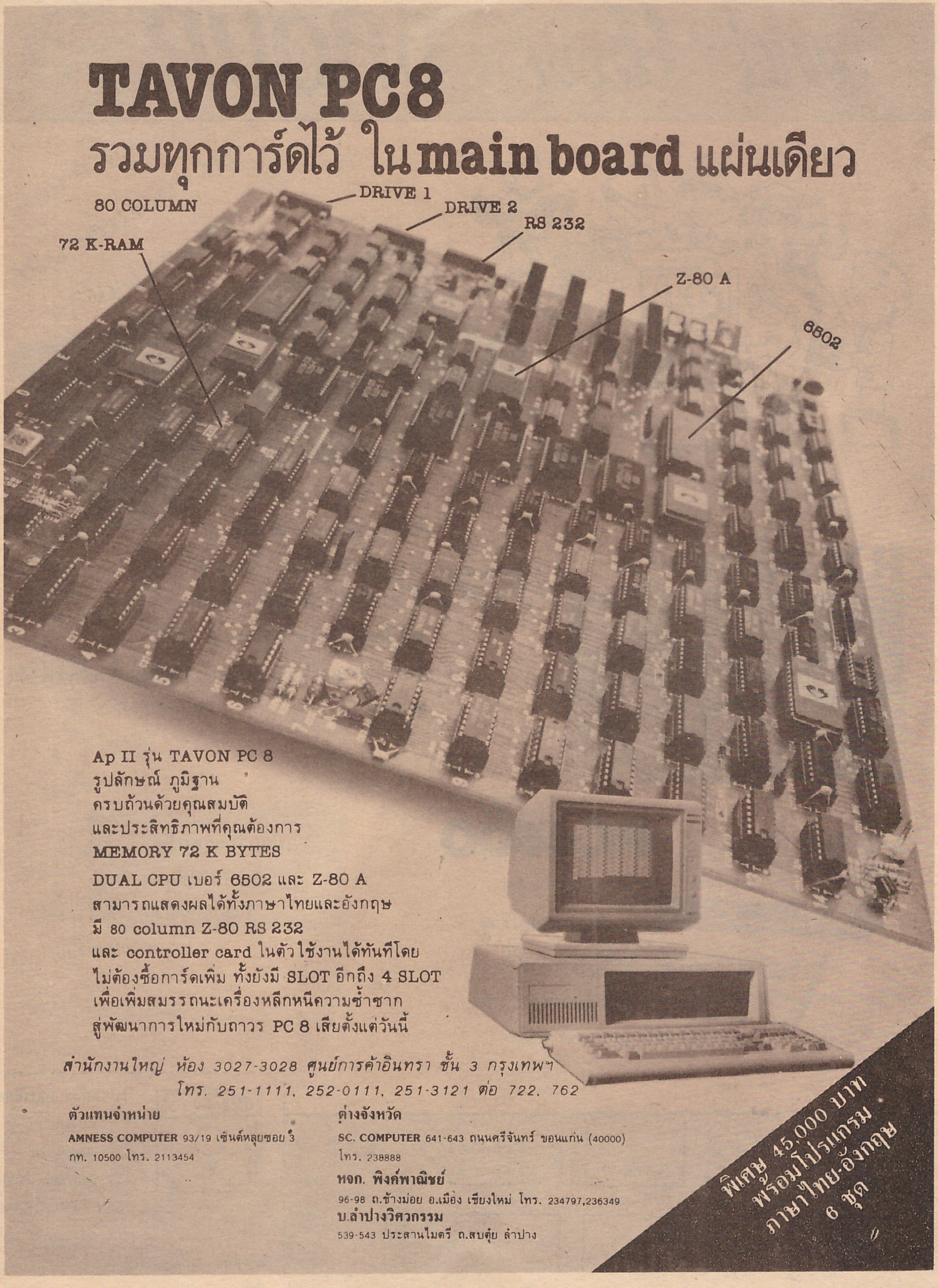


โปรดส่งรายละเอียด

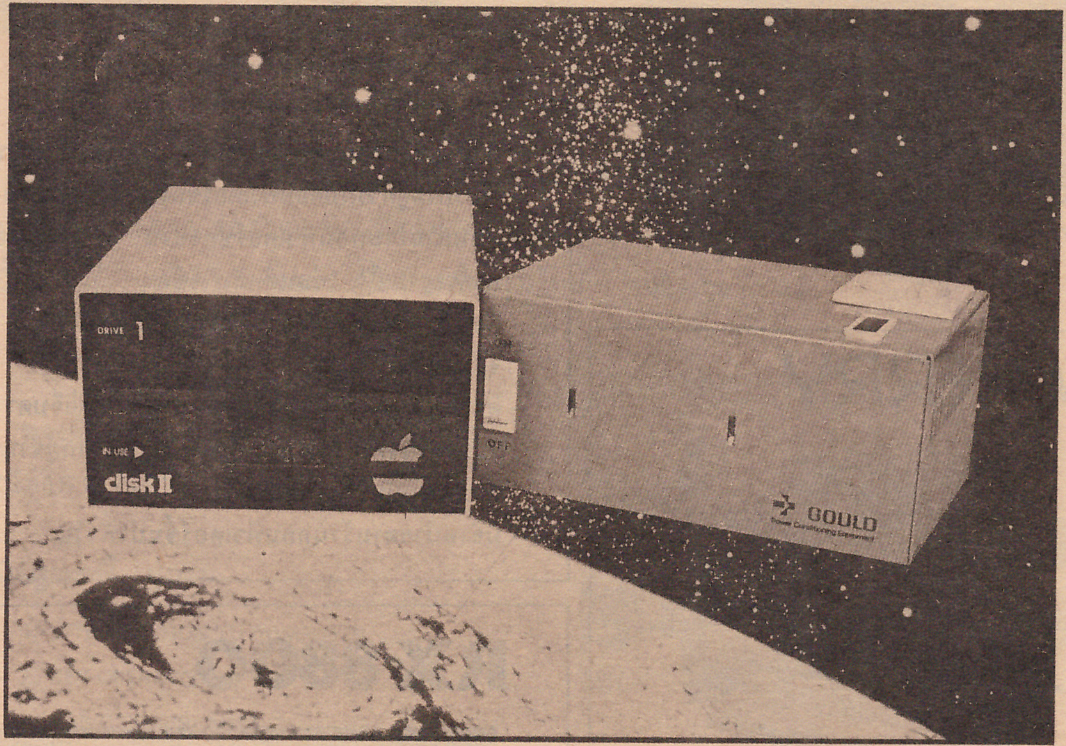
☐ สมาชิก
☐ Hardware
☐ Software และบัตร

COPY PROGRAM ฟรี

ชื่อ...
นามสกุล...
ที่อยู่...
.... โทร ....


TAVON PC8
รวมทุกการ์ดไว้ ใน main board แผ่นเดียว
80 COLUMN
DRIVE 1
DRIVE 2
RS 232
72 K-RAM
Z-80 A
6502
Ap II รุ่น TAVON PC 8
รูปลักษณ์ ภูมิฐาน
ครบถ้วนด้วยคุณสมบัติ
และประสิทธิภาพที่คุณต้องการ
MEMORY 72 K BYTES
DUAL CPU เบอร์ 6502 และ Z-80 A
สามารถแสดงผลได้ทั้งภาษาไทยและอังกฤษ
มี 80 column Z-80 RS 232
และ controller card ในตัวใช้งานได้ทันทีโดย
ไม่ต้องซื้อการ์ดเพิ่ม ทั้งยังมี SLOT อีกถึง 4 SLOT
เพื่อเพิ่มสมรรถนะเครื่องหลีกหนีความซ้ำซาก
สู่พัฒนาการใหม่กับถาวร PC 8 เสียตั้งแต่วันนี้
สำนักงานใหญ่ ห้อง 3027-3028 ศูนย์การค้าอินทรา ชั้น 3 กรุงเทพฯ
โทร. 251-1111, 252-0111, 251-3121 ต่อ 722, 762
ตัวแทนจำหน่าย
AMNESS COMPUTER 93/19 เซ็นต์หลุยซอย 3
กท. 10500 โทร. 2113454
ต่างจังหวัด
SC. COMPUTER 641-643 ถนนศรีจันทร์ ขอนแก่น (40000)
โทร. 238888
หจก. พิงค์พาณิชย์
96-98 ถ.ช้างม่อย อ.เมือง เชียงใหม่ โทร. 234797,236349
บ.ลำปางวิศวกรรม
539-543 ประสานไมตรี ถ.สบตุ๋ย ลำปาง
พิเศษ 45,000 บาท
พร้อมโปรแกรม
ภาษาไทย-อังกฤษ
6 ชุด

เหนือกว่าด้วยเทคนิค คุณภาพ และการออกแบบ



Gould คุณภาพสูงค่า คุ้มราคา

Line Conditioner ของ Gould กว่า 500 เครื่อง ได้รักษาเครื่องคอมพิวเตอร์ เครื่องมืออิเล็คโทรนิคส์ เครื่องมือวัด ไมโครคอมพิวเตอร์ เครื่องเสียง เครื่องควบคุมของระบบอุตสาหกรรมไว้ มูลค่ากว่าร้อยล้าน บัดนี้เราขอเสนอรุ่นประหยัด ราคาย่อมเยา เพียงไม่กี่พันบาท ท่านก็เป็นเจ้าของได้ เพื่อรักษาไมโคร ทีวี วีดีโอ ดิสไดรพ์

เพราะเรารักษาระดับไฟให้คงที่ ตัดสัญญาณ Transient spike และมี Overload & Over temperature ในตัว ด้วยมาตรฐานสูงของ Gould เราเคยมีประวัติการใช้งานในหม้อแปลงควบคุมแรงดันไฟแบบนี้ กว่า 200,000 ชั่วโมงแล้ว นั่นเป็นเหตุผลที่หลายคนนิยมใช้ Gould

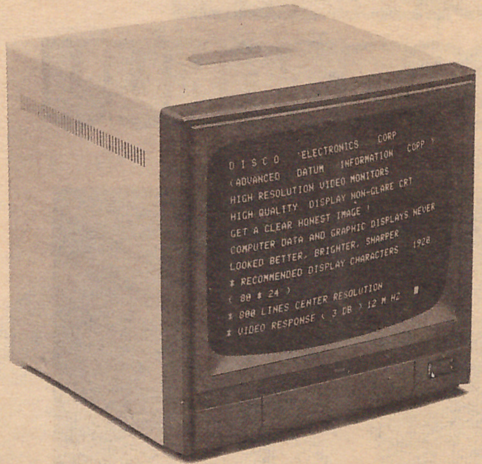
แล้วท่านละ...!

GOULD
Electronics

สนใจติดต่อ:
บริษัท แมสเวอลด์ จำกัด
MASSWORLD CO.,LTD.
30 ซ.ศึกษาวิทยา ถ.สีลม กทม.10500 โทร.2349330, 2359979

MASSWORLD 050684

DISCO UL®



ดิสโก้มอนิเตอร์ มอนิเตอร์มาตรฐาน UL

- ดิสโก้มอนิเตอร์ คือมอนิเตอร์ไม่สะท้อนแสงสำหรับ APPLE II และคอมพิวเตอร์ทั่วไป
- ดิสโก้มอนิเตอร์เหนือกว่ามอนิเตอร์อื่น เพราะเป็นหนึ่งเดียวของไต้หวันที่ UL ให้การรับรอง

  UL คือหน่วยงานตรวจสอบมาตราฐานสินค้าอิเลคทรอนิคส์ที่นำเข้าอเมริกา

  UL มีหน้าที่ตรวจสอบคุณภาพและความปลอดภัยของสินค้า

  UL อนุมัติให้สินค้าที่ตรวจสอบผ่านเท่านั้น จึงจำหน่าย ในตลาดอเมริกาได้

- เราจึงภูมิใจเสนอ ดิสโก้มอนิเตอร์ มอนิเตอร์คุณภาพอเมริกาที่คู่ควรกับคอมพิวเตอร์ของท่านในราคาไทย ๆ
- สอบถามร้านคอมพิวเตอร์ของท่าน ถึงมอนิเตอร์มาตราฐาน UL ก่อนตัดสินใจซื้อ
- ดิสโก้มอนิเตอร์ จอ 12" มาตราฐานใหญ่กว่าจอมอนิเตอร์อื่น ๆ
- ดิสโก้มอนิเตอร์ ตัดแสงสะท้อน ดูแล้วไม่เจ็บตา
- โปรดตรวจสอบใบรับรองคุณภาพ UL จากผู้จำหน่ายเพื่อความมั่นใจในคุณภาพ
- มีจำหน่ายตามร้านคอมพิวเตอร์ชั้นนำทั่วไป

SHERRY



RX-8800





PC-5500

ผู้แทนจำหน่ายแต่ผู้เดียวในประเทศไทย



บริษัท ซุปพีเรียร์ คอมพิวเตอร์ จำกัด

ชั้น 2 โรงภาพยนต์ลิโด้ เลขที่ 258/18 สยามสแควร์ซอย 3 กท.10500 โทร. 251-7078

ปิดเทอมนี้
เรียนคอมพิวเตอร์ที่
BCC คุ้มค่ากว่า

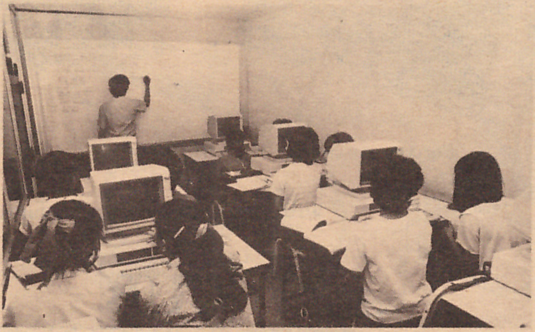
BCC
The Business Computer Centre

ศูนย์อบรมคอมพิวเตอร์ธุรกิจ

ในความควบคุมของกระทรวงศึกษาธิการ

โทร.278-5370, 278-1408

บัดนี้ BCC ล้ำหน้าในวงการโรงเรียนสอนคอมพิวเตอร์อีกครั้งด้วยการนำเอาเครื่องไมโคร-คอมพิวเตอร์ APPLE II รุ่นล่าสุด คือ IBM look alike มาใช้ในการเรียนการสอนแทนเครื่องชุดเก่าทั้งหมด ทั้งนี้เพื่อให้ผู้เรียนได้เรียนรู้และสัมผัสกับเครื่องรุ่นใหม่ ๆ ตลอดเวลา



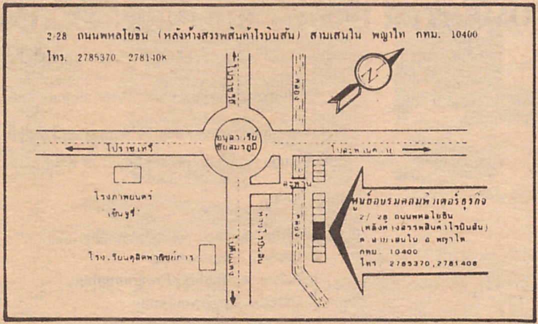
ทางด้านหลักสูตร ได้ปรับปรุงและเพิ่มเติมเพื่อประสิทธิภาพในการทำงาน เช่น...

หลักสูตรพื้นฐาน

1. INTRODUCTION TO COMPUTER
2. KEY OPERATOR
3. BASIC PROGRAMMING

หลักสูตรชั้นสูง

1. ADVANCED BASIC
2. COBOL, ADVANCED COBOL
3. CP/M, DBASE II, SUPERCALC, WORDSTAR
4. การใช้คอมพิวเตอร์ในงานบัญชี สต๊อค



ทุกหลักสูตร แจกตำราเรียนมาตรฐาน พร้อมทั้งมีประกาศนียบัตรรับรองโดยกระทรวงศึกษาธิการ

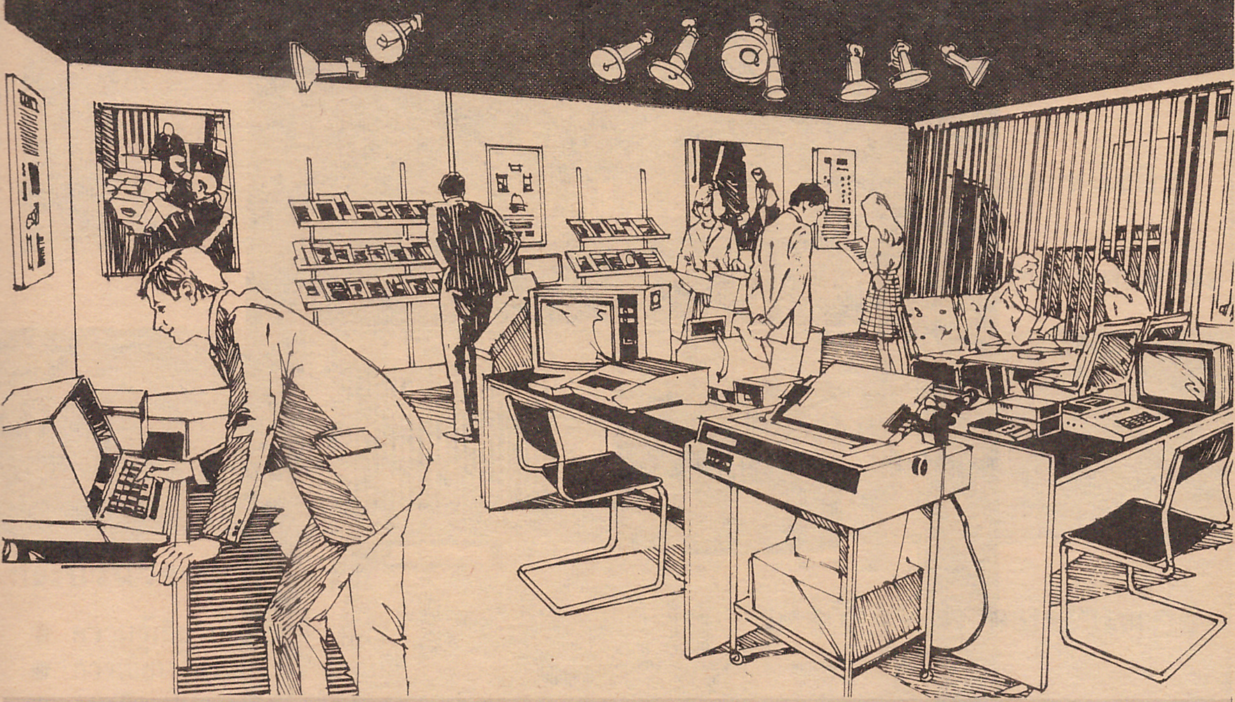
กำลังเปิดรับสมัครรุ่นใหม่! รับสมัครด่วน รับจำนวนจำกัดเพื่อการเรียนที่ได้ผล

จองที่เรียนโทร.278-5370, 278-1408 หรือที่
2/28 หลังโรบินสัน อนุสาวรีย์ชัยฯ ถนนพหลโยธิน เขตพญาไท กทม.10400

สนใจ *Micro Computer ?*

ปรึกษาผู้ที่ไว้ใจได้

data computer inc.



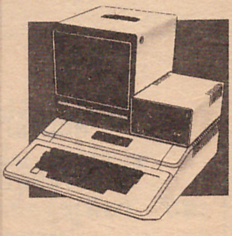
รายการสินค้าที่น่าสนใจ

- **ACCOUNTING PLUS :** เป็นโปรแกรมชุดทำบัญชี สำหรับงานธุรกิจทุกประเภท ประกอบด้วย บัญชีแยกประเภท (General Ledger), บัญชีลูกหนี้ (Account Receiveable), บัญชีเจ้าหนี้ (Account Payable), บัญชีสินค้าคงคลังเป็นโปรแกรมทางบัญชีที่ใช้ง่ายที่สุดในตลาดขณะนี้ ผู้ไม่มีความรู้ทาง Computer ก็สามารถใช้ได้อย่างมีประสิทธิภาพ เหมาะสำหรับ บริษัท, ห้างฯ ร้าน และธุรกิจอื่นๆ ทุกระดับ
- **SAM CARD** ถ้าคุณเบื่อนั่งดู MONITOR เงียบๆ SAM CARD จะช่วยให้ APPLE/APPLE COMPATBLE ของคุณพูดกับคุณได้
- **PROCESSOR II, GREEN MONITOR, DISK DRIVE, PRINTER, Z-80 CARD, 80 COLUMNS CARD, 16 K RAM CARD, PAL CARD, JOY STICK, BLANK DISKETTE** และอะไรที่เกี่ยวกับ **MICRO COMPUTER** ของคุณ 'เรามี'
- DATA COMPUTER รับประกอบ และซ่อม MICRO COMPUTER ทุกชนิด และเฉพาะ APPLE/APPLE COMPATIBLE เรามีอุปกรณ์ทุกชิ้น รวมทั้งกล่องพลาสติกจำหน่ายในราคาถูก

DATA COMPUTER INC. ☎ 251 5318

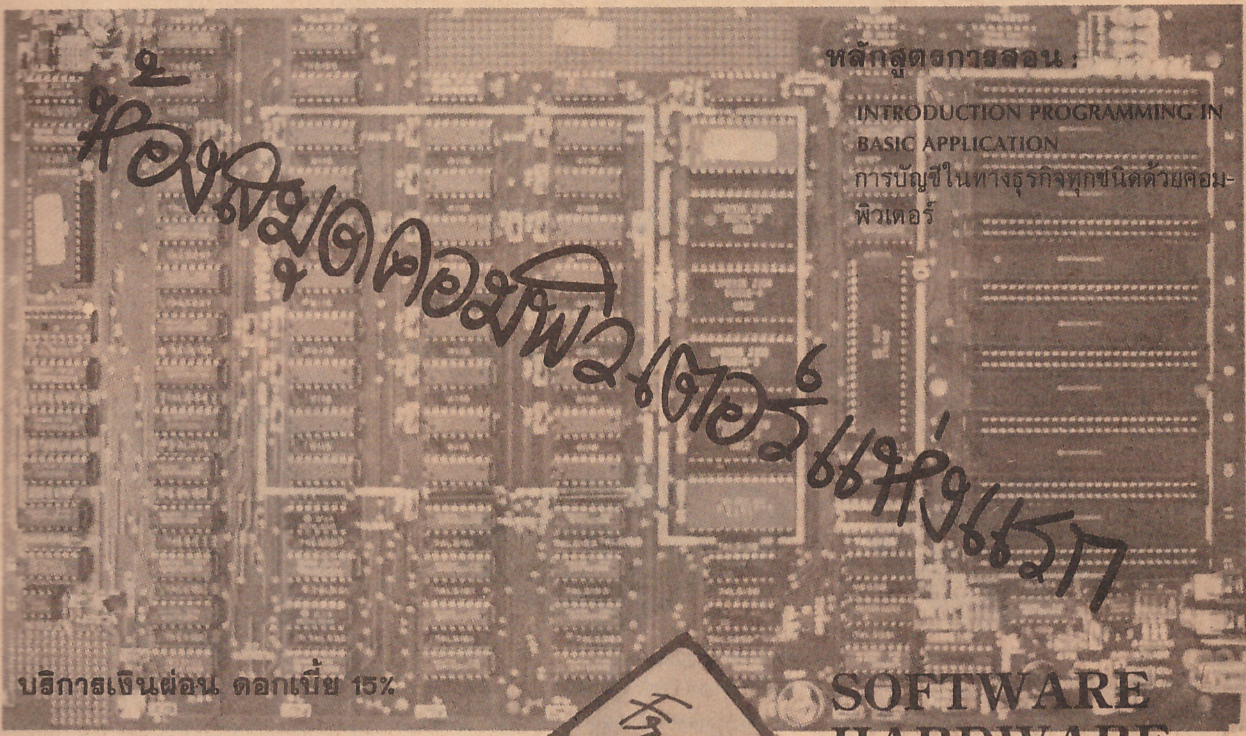
1549/6 ชั้น 2 เพชรบุรีตัดใหม่ ศูนย์การค้าออสการ์





ถ้าคุณสนใจ MICRO COMPUTER และสงสัยว่า MICRO COMPUTER จะช่วยธุรกิจของคุณได้อย่างไรบ้าง หรือคุณอยากจะศึกษา MICRO-COMPUTER เพื่อธุรกิจและความบันเทิง เชิญปรึกษากับเรา **DATA COMPUTER** ได้ในฐานะนักวิชาการ และผู้จำหน่าย MICRO COM--PUTER โดยไม่มีข้อผูกมัดใดๆทั้งสิ้น **DATA COMPUTER** จะสาทิตให้คุณเห็นว่า MICRO COMPUTER ช่วยงานธุรกิจของคุณได้อย่างไร ด้วยโปรแกรมต่างๆมากมาย และหรือด้วยโปรแกรมที่เขียนขึ้นสำหรับ 'คุณ' โดยเฉพาะ ไม่ว่าคุณจะทำธุรกิจอะไร คุ้นเคยกับ COMPUTER แค่ไหน **DATA COMPUTER** ช่วยจัดระบบ และจัดโปรแกรมให้คุณได้เสมอ ในราคาที่น่าสนใจ





ห้องสมุดคอมพิวเตอร์แห่งแรก

หลักสูตรการสอน :

INTRODUCTION PROGRAMMING IN
BASIC APPLICATION
การบัญชีในทางธุรกิจทุกชนิดด้วยคอม-
พิวเตอร์

บริการเงินผ่อน ดอกเบี้ย 15%

SOFTWARE
HARDWARE
SOFTWARE
HARDWARE
SOFTWARE
HARDWARE

จำหน่ายแผ่นโปรแกรม และเทปโปรแกรม

**สมาชิกที่ต่ออายุ ขอรับแจก
เสื้อยืดฟรี 1 ตัว**



สำหรับสมาชิก
พิเศษสุด

บริการ ใช้เครื่องคอมพิวเตอร์และ
ห้องสมุด แลกเปลี่ยนโปรแกรม ฟรี

**แจกวารสารไมโครอิเลคทรอนิคส์
กระดาษปอนด์ ฟรี**

บริการยืมเทป วี.ดี.โอ. การสอน
ส่วนลดจากการซื้อหนังสือที่ร้าน
THE BOOK CHEST สยามสแควร์
ซอย 2 10% ทุกชนิดเว้นวารสาร

สนใจ

เชิญติดต่อรายละเอียดได้ที่

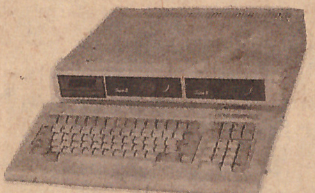
ชมรมผู้ใช้
ไมโครคอมพิวเตอร์

**236/5 ชั้น 3 โรงภาพยนต์ลิโด
(ด้านหน้าขวามือ) ปทุมวัน
กรุงเทพฯ 10500**

TEL.2523695

CREDIT COMPUTER

236/11 ชั้น 2 โรงภาพยนตร์ลิโด (ด้านหน้าขวามือ) สยามสแควร์ซอย 2 ปทุมวัน โทร.2528780



บริการเงินผ่อน ดอกเบี้ยพิเศษ

จำหน่าย

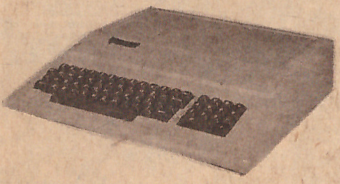
- ไมโครคอมพิวเตอร์ 8 บิท และ 16 บิท
- หนังสือคอมพิวเตอร์

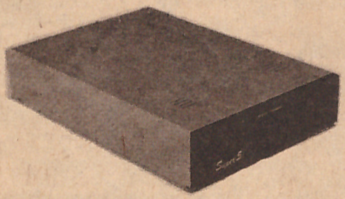
ขายส่ง และขายปลีก

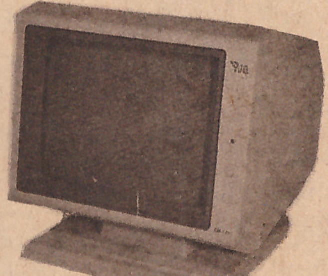
- แผ่นดิสเก็ตและเทปโปรแกรม
- มีโปรแกรมภาษาไทยเฉพาะงานจำหน่าย
- โปรแกรมวรรณึก ทุก version

ไมโครคอมพิวเตอร์
รุ่น PC-9900, PC-5500
ใช้มาตรฐานแผ่นวงจรหลัก ผลิตจากประเทศญี่ปุ่น แป้นกดข้อมูล แบบแยกส่วนจากตัวเครื่อง จำนวน 95 คีย์ มีฟังก์ชั่นคีย์ 12 คีย์ 24 ฟังก์ชั่น ทั้ง CP/M และ BASIC

ขณะนี้จนถึงสิ้นเดือนสิงหาคม 2527
สมนาคุณทุกท่านด้วย **แผ่นดิสเก็ตฟรี 1 กล่อง**





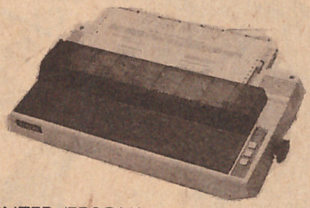


ไมโครคอมพิวเตอร์
ใช้มาตรฐานวงจรหลัก ผลิตจากประเทศญี่ปุ่น แป้นกดข้อมูล จำนวน 68 คีย์ มีแป้นคีย์ตัวเลขและฟังก์ชั่นพิเศษ 71 ฟังก์ชั่น กำหนดฟังก์ชั่นพิเศษได้อีก 8 ฟังก์ชั่น สามารถพิมพ์ซ้ำโดยอัตโนมัติ (Auto Repeat) และพิมพ์ตัวอักษรใหญ่-เล็กได้

'SUPER-5' (Slim Type)
Mini Floppy Disk Drive
เครื่องอ่านและบันทึกข้อมูลลงบนแผ่นบันทึกข้อมูล (Diskette) ขนาด 5 1/4 นิ้ว ใช้ระบบขับเคลื่อนโดยตรง (Direct Drive)

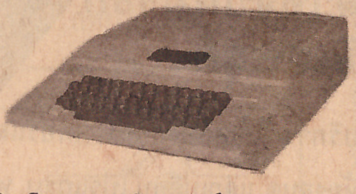
จอภาพ YJE รุ่น GM-1211
ขนาดจอภาพ 12 นิ้ว มีฐานตั้งปรับมุมก้มเงยได้ 30 องศา ปรับหมุนซ้ายขวาได้ 60 องศา แสดงภาพสีเขียว (Green) หรือสีเหลืองอำพัน (Amber) ความละเอียดของภาพ (Resolution) 1000 เส้น ช่วงกว้างของสัญญาณ (Band Width) 20 MHZ ใช้ไฟฟ้า 220 โวลท์ 50 เฮิร์ทซ์ 30 วัตต์





PRINTER 'EPSON'
เครื่องพิมพ์คอมพิวเตอร์ ใช้ได้ทั้งกระดาษธรรมดาและกระดาษคอมพิวเตอร์ พิมพ์ได้ทั้งสองทิศทาง

รุ่น FX-80	ความเร็ว 160 ตัวอักษรต่อวินาที หน้ากระดาษกว้างสุด 10 นิ้ว
รุ่น MX-100	ความเร็ว 100 ตัวอักษรต่อวินาที หน้ากระดาษกว้างสุด 16 นิ้ว
รุ่น FX-100	ความเร็ว 160 ตัวอักษรต่อวินาที หน้ากระดาษกว้างสุด 16 นิ้ว



16 บิท ไมโครคอมพิวเตอร์
หน่วยความจำ 128 ขยายได้ถึง 512 ใช้ MS–DOS ใช้โปรแกรมของ IBM PC มี HARD DISK (10 M bytes)

ไมโครคอมพิวเตอร์
ใช้มาตรฐานแผ่นวงจรหลัก ผลิตจากประเทศญี่ปุ่น แป้นกดข้อมูล จำนวน 53 คีย์ มีฟังก์ชั่นพิเศษ

หากท่านสนใจเรียนคอมพิวเตอร์ ที่ค่าใช้จ่ายถูกที่สุด สามารถนำไปใช้ในการศึกษาต่อหรือเพื่อทำงาน

โปรดกรอกรายละเอียดข้างท้ายนี้ ท่านจะได้รับรายละเอียดการศึกษาฟรี

สนใจศึกษา ☐ 8 บิท ☐ 16 บิท

การเรียนแบบ ☐ ตัวต่อตัว ☐ ทางไปรษณีย์ ☐ กลุ่มย่อย

ค่าอบรม ☐ ไม่เกิน 300.- ☐ ไม่เกิน 500.- ☐ ไม่เกิน 1,000.

ชื่อ-นามสกุล............................ ที่อยู่............................ โทร.................

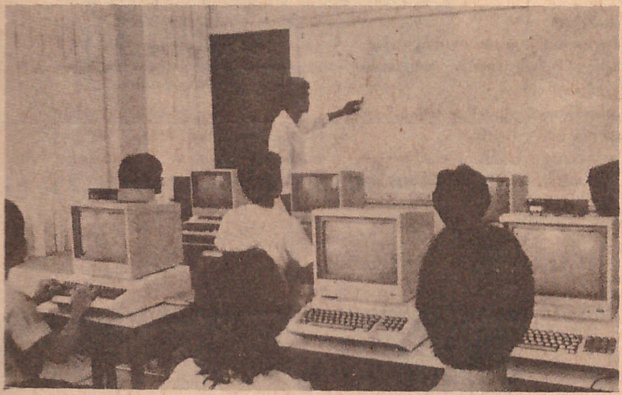
ศูนย์พัฒนาการศึกษาคอมพิวเตอร์

75/2 ชั้น 2 รามคำแหง 35 ถ.รามคำแหง หัวหมาก บางกะปิ กทม. 10240

โทร. 3145727

สอนคอมพิวเตอร์

ขั้นพื้นฐาน-โปรแกรมระดับสูง



หลักสูตร

1. การเขียนโปรแกรมภาษาเบสิค

- INTRODUCTION TO COMPUTER
- KEY OPERATOR
- BASIC PROGRAMMING
  - APPLESOFT BASIC
  - CP/M (MBASIC), DOS
  - FILES, SEQUENTIAL, RANDOM

(สอนทุกหัวข้อในชั้นเดียว ราคาถูกกว่าที่อื่น)

2. APPLICATION PACKAGE

- CP/M, SUPERCALC
- dBASE II, WORDSTAR

☐ ดำเนินการสอนโดยอาจารย์จากจุฬาฯ และผู้ทรงคุณวุฒิ ด้านคอมพิวเตอร์

☐ แจกตำราฉบับสมบูรณ์เขียนโดยคณาจารย์จุฬาฯ และ DISKETTE PROGRAM

☐ มีห้องคอมพิวเตอร์สำหรับสมาชิกฝึกใช้นอกเวลาเรียนและหลังจากจบ ฟรี

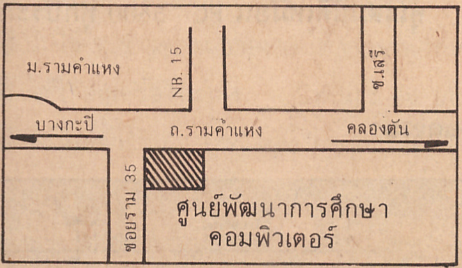
☐ บริการให้เช่าเครื่องฝึกโปรแกรม (สำหรับบุคคลภายนอก)

วิธีเรียน

แบ่งการเรียนเป็น 2 แบบ

1. สมัครและเข้าเรียนวัน, เวลาใดก็ได้ภายในเวลา 9.00-15.00 น. จันทร์-ศุกร์
2. สมัครและเรียนเป็น CLASS มีรอบวันและเวลาดังนี้

● จันทร์-ศุกร์	15.00-17.00 น.
● จันทร์-ศุกร์	17.00-18.30 น.
● จันทร์-พุธ-ศุกร์	18.30-20.30 น.
● อังคาร-พฤหัสฯ	18.30-20.30 น.
● เฉพาะเสาร์หรืออาทิตย์	



เปิดสอนรุ่นใหม่ทุกเดือน (รับจำนวนจำกัด)

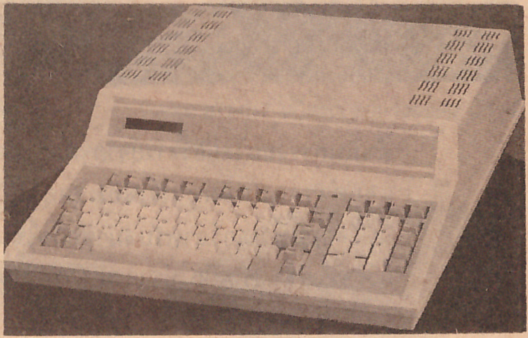
PC - 5500

11,000.-

- 64 K PERSONAL COMPUTER
- 100% EXPANDABLE UP TO 256 K RAM
- 10 USER DEFINE KEY (BATTERY BACK UP)
- 84 KEYS 188 FUNCTION KEY:
  - 94 BASIC FUNCTIONS
  - 94 CP/M FUNCTIONS
- AUTO REPEAT KEY ทุก KEY
- AUTO LINE NUMBER (10,20,30)
- FUNCTION CONTROL KEY
- CAGE พลาสติคล้วน สีนุ่มนวลและเป็น FURNI TURE ได้ในตัว
- ฝาเปิดขึ้นด้านบน สามารถถอดหรือเสียบ INTERFACE CARD ต่าง ๆ ได้ง่าย การตรวจซ่อม เมนบอร์ดทำได้สะดวกที่สุด
- ใช้ MAIN BOARD ซึ่งมีคุณสมบัติเทียบเท่า APPLE และเล่นโปรแกรมของ APPLE II และ CP/M ได้ทุกโปรแกรม
- สามารถเพิ่ม ROM เป็น 24 K ROM มีภาษา FORTH 79 เพิ่มอีก 1 ภาษา
- ทุกรุ่นสามารถใส่ DISK DRIVE เข้าไว้ในเครื่องได้ 2 เครื่อง มีแป้นสำหรับติดพัดลมระบายอากาศอยู่ในกล่องเรียบร้อย
- สามารถเลือกใช้ KEY BOARD รุ่นอื่น ๆ ในท้องตลาดได้ทุกรุ่น ทุกแบบ
- สามารถขยายหน่วยความจำในการ RUN โปรแกรม VISI CALC จาก 34 K เป็น 145 K เมื่อใช้แผงวงจร 128 K RAM CARD 1 แผง และสามารถ SAVE FILE ที่มีความยาวมาก ๆ ลงแผ่น DISKETTE ได้มากกว่า 1 แผ่น



PC - 5500 ใช้ KEY **แบบแยกส่วน รุ่น** KB -2000



RX - 8800 รุ่นพิเศษ

คุณสมบัติเหมือน PC - 5500 ทุกประการ

บริษัทแถมโปรแกรมทุกโปรแกรม ฟรี ตลอด

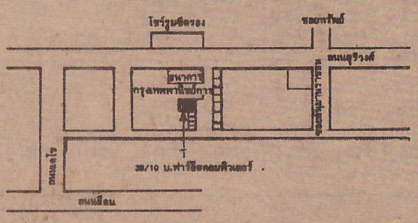
มีโปรแกรม UTILITY **ต่าง ๆ แถมฟรี**

เช่น : LOCK SMITH VERSION 5.0
: VIDEX VIDEO TERM UTILITY
: BACK OF TRICK
: NIBBLE AWAY II รุ่นใหม่

ผู้ที่สนใจโปรแกรมภาษาไทย โปรดติดต่อสอบถามได้ตลอดเวลา

บริษัท ฟาร์อีสท์ คอมพิวเตอร์ เอ็นยิเนียริ่ง จำกัด

38/10 ซอยอนุมานราชธน
ถนน สุริวงศ์ บางรัก กทม. 10500
โทร. 235-9230-1



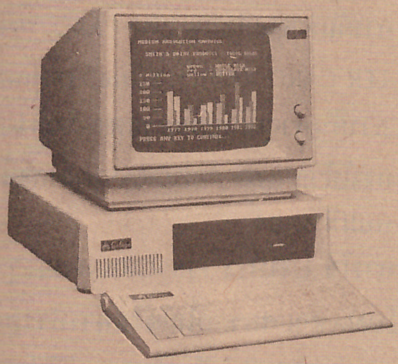
แนะนำอุปกรณ์
สำหรับเครื่องไมโครคอมพิวเตอร์
เพื่อใช้งานด้านภาษาไทยได้สะดวกทุกโปรแกรม

หยุดความสับสนในการเลือก
ซื้อไมโครฯ ของท่าน

โปรดพิจารณาข้อเสนอของเราก่อน

เนื่องจากโปรแกรมภาษาไทยสำหรับไมโครฯ APPLE COMPATIBLE หลายโปรแกรม ใช้ CODE ภาษาไทยภายในเครื่องไม่ตรงกัน เราได้พัฒนา 80 COLUMNS CARD ให้แสดงตัวอักษรไทยสำหรับทุกโปรแกรมโดยไม่ต้องถอดสับเปลี่ยน ROM CHARACTER ภาษาไทย และยังคงมีอักษรอังกฤษทั้งธรรมดาและ INVERSE อยู่ เหมือนเดิม

- ลูกค้าของบริษัทฯ ที่ซื้อเครื่องไมโครฯ จากเราจะได้รับการบริการฟรี
- รับดัดแปลงให้กับลูกค้าทั่วไปในราคาพิเศษ 400 บาท



บริษัทจำหน่าย PRINTER ในราคาปกติพร้อมทั้ง ROM ภาษาไทย สำหรับโปรแกรมภาษาไทยทุกโปรแกรม และดัดแปลงให้ใช้ได้โดยไม่ต้องถอดสับเปลี่ยน ROM ภายในเครื่องแต่อย่างใด

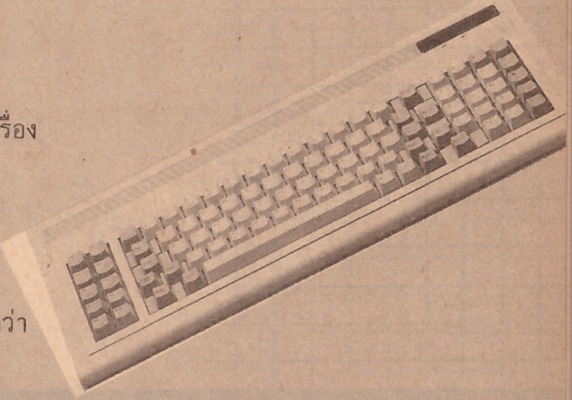
- PRINTER 'SUPER 5' CP 80 พร้อม ROM ภาษาทุกโปรแกรมดัดแปลงเรียบร้อยแล้ว ราคาเครื่องละ 13,500 บาท
- PRINTER 'ADMATE' พร้อม ROM ภาษาไทยทุกโปรแกรม ดัดแปลงเรียบร้อยแล้ว ราคาเครื่องละ 14,500 บาท

เครื่องไมโครฯ APPLE COMPATIBLE ของบริษัทฯ สามารถดัดแปลง MAIN BOARD ให้มีภาษา FORTH 79 อยู่ในตัวโดยใช้ ROM ขนาด 24 K ROM และไม่ต้อง ใช้ SOFTWARE แต่ประการใด

- ลูกค้าของบริษัทจะได้รับการบริการฟรี
- รับดัดแปลงให้กับลูกค้าทั่วไปในราคาพิเศษ 600 บาท

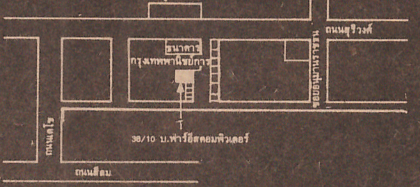
บริษัทฯ ขอเสนอ KEY BOARD รุ่นพิเศษ 'KB 2000' เป็นแบบเดียวกับที่ใช้กับเครื่องไมโครฯ 16 บิททั่ว ๆ ไป รูปร่างเหมือนกับของ IBM

- อายุการใช้งานของแต่ละแป้นอักษรไม่ต่ำกว่า 20 ล้านครั้ง
- ไม่มีอาการอักษรตกหล่นหรือกระโดดซ้ำกันอย่างเด็ดขาด
- ไม่ใช้ SILICONE RUBBER แต่ใช้ CONTACT แบบพิเศษ กดได้นิ่มนวลกว่า
- ใช้ได้กับเครื่องไมโครฯ 8 บิท และ 16 บิททั่ว ๆ ไป
- มีสีสันเข้ากับ CASE PC-5500 และ CHERRY รุ่นใหม่



บริษัท ฟาร์อีสท์ คอมพิวเตอร์ เอ็นยิเนียริ่ง จำกัด

38/10 ซอยอนุมานราชธน
ถนน สุริวงศ์ บางรัก กทม. 10500
โทร. 235-9230-1



ฟาร์อีส คอมพิวเตอร์ เอ็นยิเนียริ่ง

โดย ไตรรงค์ ตั้นตระกูล (กรรมการผู้จัดการ)

ได้เล็งเห็นความขาดแคลนช่างฝีมือในการซ่อม แก้ไข ดัดแปลงเครื่องไม
จึงได้ร่วมกันจัด **สัมมนาอบรม ช่างฝีมือ** ในหลักสูตรซ่อมเครื่อ
เน้นเฉพาะไมโครฯ รุ่น APPLE COMPATIBLE ทุกรุ่นทุกแบบ

หลักสูตรที่อบรม :

- ระบบการทำงานของเครื่องไมโครฯ แบบ
- การค้นหาจุดเสียบน MAIN BOARD แล
- การค้นหาจุดเสียบน Z-80 CARD, 16 K RAM LER CARD และการซ่อมแก้ไขทุกอาการ
- การค้นหาจุดเสียและการซ่อมแก้ไข PO
- การใช้ UTILITY PROGRAM ต่าง ๆ เพื่อช่วย คอมพิวเตอร์
- การใช้ UTILITY PROGRAM ต่าง ๆ เช่น AWAY II ฯลฯ เพื่อใช้ในการทำ BACK **COPY** ทั่ว ๆ ไป ซึ่งเน้นเฉพาะโปรแกรม
- การดัดแปลงระบบภาษาไทยให้ใช้ได้กับ

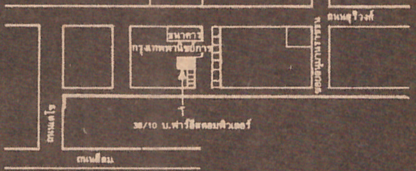
ดำเนินการสอนและอบรมโดย คุณไตรรงค์ ตั้นตระกูล และ
อบรมด้วยหลักสูตร หลักวิธีการตรวจสอบคุณภาพไมโครคอม
และมีอาจารย์จากต่างประเทศร่วมอำนวยการบรรยายอบรมในหลักสูตร
ผู้ที่ผ่านการสัมมนาอบรมแล้ว รับรองว่าสามารถจะประกอบ
ได้อย่างแน่นอน !

เปิดสัมมนาอบรมเพีย

เปิดรับลงทะเบียนตั้งแต่บัดนี้เป็นต้นไป
ระยะเวลาการอบรม 1 เดือน (90 ชั่วโมง) ทุกวัน
ติดต่อสอบถามรายละเอียดและสมัครได้ทั้งสองแห่ง ตลอดเวลา

บริษัท ฟาร์อีสท์ คอมพิวเตอร์ เอ็นยีเนียริ่ง จำกัด

38/10 ซอยอนุมานราชธน
ถนน สุริวงศ์ บางรัก กทม. 10500
โทร. 235-9230-1



และที่

ริงและ คอสมิค คอมพิวเตอร์

โดย ดิศักดิ์ ศรีสุวรรณ (กรรมการผู้จัดการ)

COSMIC

ไม โครคอมพิวเตอร์ในตลาดคอมพิวเตอร์เมืองไทยที่กำลังตื่นตัว
ครื่ ไมโครคอมพิวเตอร์ และอุปกรณ์ทุกชนิดที่ใช้ร่วมกับเครื่องไมโครคอมพิวเตอร์ โดย

แบบ APPLE COMPATIBLE ทุกแบบ
แล ซ่อม MAIN BOARD แบบรวดเร็ว
RA CARD, 80 COLUMNS CARD, DISK DRIVE CONTROL–
การ
PO WER SUPPLY ทุกรุ่นทุกแบบ
ช่วย ประหยัดเวลาในการตรวจหาจุดบกพร่องของระบบไมโคร-

น LOCK SMITH VERSION 5, BACK OF TRICK, NIBBLE
CK UP โปรแกรมที่ไม่สามารถ COPY ด้วย**โปรแกรม**
รม ภาษาไทยที่มีราคาแพงและไม่มีแผ่นสำรอง (BACK UP)
บ เครื่อง AP II รุ่นต่าง ๆ

ละ คุณดิศักดิ์ ศรีสุวรรณ ตลอดจนทีมอาจารย์ผู้ร่วมงานอีกหลายคน
คอม พิวเตอร์ ในอุตสาหกรรมการผลิตเครื่องไมโครคอมพิวเตอร์ในต่างประเทศ
สูตร ที่สำคัญ ๆ
อบ อาชีพในด้านการให้บริการทางด้านฮาร์ดแวร์ของระบบไมโครคอมพิวเตอร์

ย รุ่นเดียวเท่านั้น

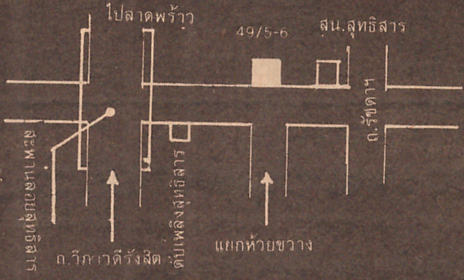
อัตราค่าลงทะเบียน 5,000 บาทต่อหนึ่งท่าน
เริ่มสัมมนาอบรมตั้งแต่วันที่ 15 กรกฎาคม 2527

ละที่

บริษัท คอสมิค คอมพิวเตอร์ จำกัด

49/5-6 ถนนสุทธิสาร ห้วยขวาง กท. 10400

โทร. 2773270



Sofpac

BUSINESS
- DATA BASE
- Word Processing
- Spread sheet
- etc.

GAMES
- Adventure
- Arcade
- etc.

EDUCATION
- Mathematic
- Science
- Language
- etc.

UTILITY
- Copy programs
- Service program
- etc.

Sofpac คือ Software package สำหรับเครื่อง Apple II และ IBMPC

Sofpac - คือโปรแกรมสำเร็จรูปพร้อมคู่มือการใช้งาน

Sofpac - มีทั้ง software ทางด้าน Business, Games Education และ Utility Program

Sofpac - มีโปรแกรมสำหรับ Apple มากกว่า 500 program

Sofpac - มีโปรแกรมสำหรับ IBMPC มากกว่า 30 program

Sofpac - มี software ใหม่ ๆ จาก USA, Hongkong และ Singapore อยู่เสมอ

Sofpac - จัดทำโดยผู้มีความรู้ความชำนาญเรื่อง software โดยเฉพาะ

Sofpac - GUARANTEE DISK ทุกแผ่น 6 เดือนเต็ม

Sofpac....Software package สำหรับ Apple II และ IBMPC

กรุณาส่งรายชื่อ Software package สำหรับเครื่อง

☐ **APPLE II** ☐ **IBMPC**

ไปให้ข้าพเจ้าตามที่อยู่ข้างล่างนี้

ชื่อ..

ที่อยู่..

....

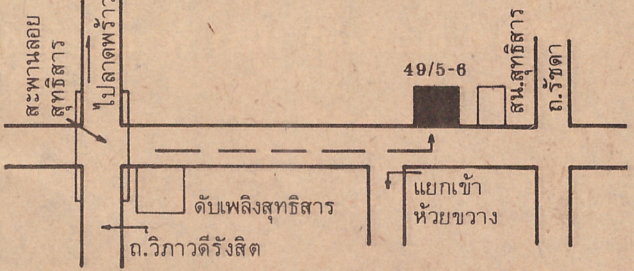
โทรศัพท์..

สนใจโปรดสอบถามรายละเอียดได้ที่

COSMIC COMPUTER CO., LTD

49/5-6 ถนนสุทธิสาร ห้วยขวาง กท.10400 โทร. 2773270

ดำเนินงานโดย ดิศักดิ์ ศรีสุวรรณ

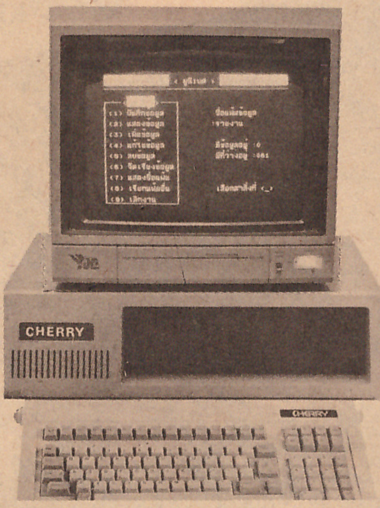


จาก APPLE สู่ CP/M มาจนถึงคอมพิวเตอร์ตระกูล CHERRY III

ไมโครคอมพิวเตอร์ที่เพียบพร้อมด้วยคุณลักษณะที่พิสูจน์ตนเองได้ นี่มิใช่คำโฆษณาแต่คุณลักษณะที่แฝงอยู่กับ CHERRY III รอการพิสูจน์จากท่าน คุณลักษณะและเอกลักษณ์นี้มีอยู่ในเครื่องคอมพิวเตอร์ขนาดใหญ่ ราคาแพง ขณะเดียวกัน CHERRY III ยังมีคุณลักษณะที่กว้างไปไกลกว่านั้น ด้วยเหตุนี้ CHERRY III จึงได้มีโอกาสรับใช้ในวงราชการ รัฐวิสาหกิจและธุรกิจเอกชนมากมาย

เริ่มจากหัวใจสำคัญของเครื่องคือ **ไมโครโปรเซสเซอร์** ไมโครโปรเซสเซอร์ที่ใช้ คือ 6502 และ Z80 สิ่งประดิษฐ์อันทรงพลังที่ทุกท่านยอมรับนี้ถูกบรรจุอยู่บนแผ่นคอมพิวเตอร์ แผ่นเดียว ด้วยการคัดเลือกอุปกรณ์และผ่านการ ตรวจสอบในขั้นตอนในการผลิตไทมิ่ง ในการทำงานของระบบวงจร LOGIC และหน่วยความจำ RAM กับไมโครโปรเซสเซอร์จะ SYNCHRONIZE กันพอดี ทำให้ไม่มีปัญหาการหยุดชะงักการทำงานของโปรแกรมในระบบ CP/M และอื่น ๆ อันเป็นปัญหาที่เกิดขึ้นกับไมโครคอมพิวเตอร์ ราคาถูกเช่น AP II หรือคอมพิวเตอร์ที่มีหน้าตาคล้ายกันเป็นต้น

หน่วยความจำขนาด 192K ของ CHERRY III นี้สามารถขยายได้สูงถึง 512K ไบท์(โดยการเพิ่ม EXTENDER RAM CARD ของบริษัท PERSONAL COMPUTER PRODUCT INC.) ซึ่งท่านจะสามารถเขียนโปรแกรมได้ซับซ้อนมากกว่าเดิม อีกทั้งยังสามารถใช้ RAM DISK ได้อีกด้วย RAM DISK นี้จะทำหน้าที่เสมือนดิสค์ไดรฟ์ตัวหนึ่ง ท่านสามารถนำข้อมูลเข้าไปใน RAM DISK เพื่อการค้นหาหรือแก้ไขข้อมูลได้อย่างรวดเร็วกว่าปกติเกือบ 10 เท่า ! นอกจากนี้ท่านยังสามารถสร้าง MODEL หรือข้อมูลขนาดใหญ่และซับซ้อนมากขึ้นใน VISICALC มากถึง 4 เท่า !



ระบบจ่ายไฟในตัว ของ CHERRY III ก็เป็นจุดเด่นที่สำคัญอย่างหนึ่ง คือมีระบบ ป้องกันอย่างดีเลิศกว่าคอมพิวเตอร์ทั่วไป คือเป็นแบบสวิชชิ่งเพาเวอร์ที่มีระบบป้องกันถึง 3 ชั้น ชั้นแรกประกอบด้วยระบบปรับแรงดันไฟโดยอัตโนมัติ ในกรณีที่เกิดไฟตกหรือไฟ เกินระบบนี้จะเพิ่มเมื่อไฟตกจะลดเมื่อไฟเกิน ทำให้คอมพิวเตอร์ปลอดภัยเหมาะสำหรับท้องถิ่น ที่มีปัญหาไฟไม่สม่ำเสมอ ชั้น 2 มีระบบป้องกันไฟกระชาก (SURGE/TRANSIENT SUPPRESSURE) เพื่อป้องกันแรงดันไฟสูงที่เพิ่มขึ้นอย่างรวดเร็ว สุดที่ฟิวส์ธรรมดาจะตัดได้ทัน เช่นไฟ สปาร์คหรือฟ้าผ่า ณ สถานที่ใกล้เคียงเป็นต้น ชั้นสุดท้ายก็คือการระบบป้องกันไฟเกิน สำหรับอุปกรณ์ไอซีต่าง ๆ บนแผ่นคอมพิวเตอร์ของท่าน สิ่งเหล่านี้เป็นคุณลักษณะส่วนหนึ่งที่ CHERRY III สามารถให้กับท่าน

ระบบภาษาไทย นับเป็นคอมพิวเตอร์อีกรุ่นหนึ่งที่มีภาษาไทยอยู่ในตัวถึง 2 ระบบ ให้ท่านเลือกใช้ตามใจชอบ ระบบภาษาไทยที่กล่าวถึงนี้คือ THAI APPLE SOFT BASIC (THAI DOS 3.3) ซึ่งแสดงผลได้ 40 ตัวอักษร/บรรทัด จำนวน 12 บรรทัด สามารถผสมภาพกราฟฟิคได้และภาษาไทยระบบ CP/M 2.2 ซึ่งแสดงผลได้ 80 ตัวอักษร/บรรทัด จำนวน 12 บรรทัด อันเป็นระบบใหม่ให้ตัวอักษรชัดเจนในการแสดงผลต่างจากที่ท่านเคยเห็นกันมาแล้วในคอมพิวเตอร์ราคาถูกทั่วไป

ความสามารถในการขยาย ท่านสามารถขยายให้เป็น 16 บิท 100% IBM-PC COMPATIBLE ได้ด้วย MPX-16 MULTIBOARD จากอเมริกาประดิษฐ์คิดค้นโดย STEVE CIACIA ซึ่งทางบริษัทได้ลิขสิทธิ์ในการนำมาติดตั้งให้กับ CHERRY III ทำให้สามารถ RUN โปรแกรมต่าง ๆ ของ IBM-PC ได้เช่น LOTUS 1-2-3 เป็นต้น นอกจากนี้ ด้วย 68000 CARD ของ SAYBROOK ทำให้ CHERRY III สามารถเป็นคอมพิวเตอร์ระบบ 32 บิทได้โดยใช้ระบบ UNIX

การให้บริการ ข้อนี้เป็นหัวใจของ CHERRY III และบริษัทฯ ซึ่งสามารถให้ท่านได้คุ้มค่ากว่าเงินทุกบาทที่ท่านลงทุน โปรดอ่านระบบ RISK FREE INVESMENT PLAN ของทางบริษัทฯ ในหน้าโฆษณาสุดท้าย ท่านจะอบอุ่นและประทับใจ

ซื้อ CHERRY III แถมโปรแกรมภาษาไทย 7 รายการ มูลค่า 12,000 บาท ฟรี !

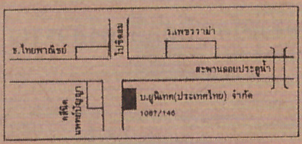
1. ระบบเวอร์ดโปรเซสซิ่งไทย (THAI WORDPROCESSOR)
2. ระบบเวอร์ดชีทไทย (THAI SUPERCALC)
3. ระบบบริหารสินค้าคงคลัง (THAI INVENTORY CONTROL)
4. ระบบจ่ายเงินเดือน (THAI PAYROLL)
5. DBASE II ไทย (ระบบใหม่)
6. MBASIC ไทย (ระบบใหม่)
7. THAI APPLESOFT (THAI DOS 3.3)

พิสูจน์ความสามารถของ CHERRY III ได้วันนี้ที่ ยูนิเทค

บริษัท ยูนิเทค (ประเทศไทย) จำกัด

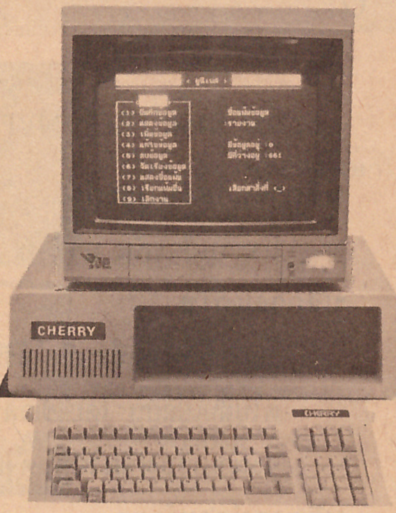
1087/146 ปากซอยจารุรัตน์ ถ.เพชรบุรี ตรงข้ามธนาคารไทยพาณิชย์

สาขาใหญ่ กรุงเทพฯ โทร. 251-6465



CHERRY III-EX คอมพิวเตอร์ที่เน้นความเร็วสูงในการทำงาน

เป็นไมโครคอมพิวเตอร์อีกรุ่นที่ทางบริษัท ยูนิเทค ภูมิใจเสนอและได้รับใช้ในหน่วยงานเอกชนมากมายและวงราชการหลายแห่งจากการที่ได้พิสูจน์ตนเองในด้านความสามารถ มีคุณลักษณะเด่น 9 ประการ ดังนี้ :



● MULTIPROCESSOR

มีไมโครโปรเซสเซอร์ 4 ตัวในเครื่องเดียวกันและแต่ละตัวทำงานกันเป็นเอกเทศ เช่นการทำงานในระบบ CP/M PLUS นี้ไม่ได้ใช้ CLOCK และหน่วยความจำของ 6502 แต่อย่างใด ทำให้ทำงานได้เร็วกว่า CP/M ปกติถึง 5 เท่า ! โดยที่ไม่มีปัญหาในการทำงานแต่อย่างใด ไมโครโปรเซสเซอร์ที่ใช้คือ Z 80 A ทำงานด้วยความเร็ว 6 MHZ นอกจากนี้ก็ยังมี 6502A และ Z 80 A อย่างละตัวที่ทำให้สามารถใช้โปรแกรมของ APPLE II+ และ CP/M2.2 ไว้

● ERGONOMIC KEYBOARD WITH FUNCTION

คีย์บอร์ดที่ใช้เป็นแบบ DETACHABLE ขนาด 95 คีย์พร้อม - FUNCTION ภาษาไทยอีกทั้งสามารถปรับระดับเอนได้ถึง 2 ตำแหน่ง มีไมโครโปรเซสเซอร์ขนาด 4 บิท ควบคุมการทำงาน

● RELIABILITY

มีระบบป้องกันถึง 3 ชั้น ซึ่งเป็นหัวใจของเครื่องที่ไม่มีในคอมพิวเตอร์รุ่นอื่น กล่าวคือตัวมันมี AUTOMATIC VOLTAGE STABILZER- ในตัวคือมันจะปรับแรงดันไฟได้เองโดยอัตโนมัติเมื่อไฟตกและไฟเกิน ชั้นที่ 2 คือมีระบบ TRANSIENT SUPPRESSURE ในตัวป้องกันไฟกระชาก เช่นไฟกระชากที่เกิดจากฟ้าผ่าหรือเครื่องชาร์จไฟอื่น ๆ ชั้นที่ 3 เป็นระบบป้องกันสำหรับอุปกรณ์ประเภท SEMICONDUCTOR เช่น อุปกรณ์ IC ต่างเป็นต้น อีกทั้งการควบคุมคุณภาพอีกครั้งหนึ่งทำให้ CHERRY III มี RELIABILITY ในการทำงานสูงยิ่ง

● THAI LANGUAGE CAPABILITY

เป็นเครื่องแรกที่มีระบบภาษาไทยในตัวถึง 2 ระบบ มีให้เลือกใช้ได้ทันทีคือ ระบบภาษาไทย APPLESOFT BASIC (THAI DOS 3.3) แสดงผลได้ ตัวอักษร/บรรทัด 12 บรรทัดและภาษาไทยในระบบ CP/M 2.2 ซึ่งแสดงผลแบบใหม่ล่าสุดคือ 80 ตัวอักษร/บรรทัด 12 บรรทัด ตัวหนังสือชัดกภาษาไทยรุ่นอื่น ๆ ในระบบ CP/M ของ APPLE

● MEMORY

หน่วยความจำมาตรฐานขนาด 320 K ไบท์ สามารถขยายได้ถึง 512 K ไบท์ โดย EXTENDER RAM CARD ของบริษัท PERSONAL COMTER PRODUCT INC. ซึ่งในส่วนของ 320 K ไบท์นี้จะถูกแบ่งนำมาใช้ระบบต่าง ๆ กันเช่น RAM DISK เป็นต้น การทำงานของ RAM DISนี้จะให้ความเร็วในการทำงานเสมือนดิสค์ไดรฟ์ตัวหนึ่ง แต่เร็วกว่าถึง 10 เท่า ! และปราศจากข้อผิดพลาด เหมาะอย่างยิ่ง สำหรับโปรแกรมที่ใช้ใการค้นหาข้อมูลมากและรวดเร็ว ซึ่งเมื่อนำ RAM DISK นี้บวกกับ CP/M PLUS จะได้ความเร็วเพิ่มขึ้นมากกว่า 10 เท่าขึ้นไป

● SOFTWARE SUPPORT

มีโปรแกรมสำเร็จรูปในปัจจุบันถึง 12,000 (ภาษาอังกฤษ)และโปรแกรมภาษาไทยทั้ง 40 คอลัมน์และ 80 คอลัมน์อย่างคับคั่ง

● EXPANDIBILITY

ด้วย MPX-16 ออกแบบโดย STEVE CIACIA ทำให้สามารถเพิ่ม 100% IBM-PC COMPATIBLE ให้กับ CHERRY III-EX รวมทั้งขยายให้เป็ 32 บิทได้โดยใช้ 68000 CARD ของ SAYBROOK

● MULTI-SYSTEM

สามารถทำงานได้ถึง 3 ระบบ คือระบบ DOS 3.3 ของ APPLE CP/M 2.2 และ CP/M PLUS ซึ่งทำงานด้วยความเร็วขนาด 6 MHZ ได้ทันทีโดยไม่ต้องดัดแปลงแก้ไขแต่อย่างใดและสามารถแสดงผลได้ดังนี้คือ ภาคภาษาอังกฤษ 40 คอลัมน์ 24 บรรทัด 80 คอลัมน์ 24 บรรทัด 70 คอลัมน์ 24 บรรทัด และ 255 คอลัมน์ 24 บรรทัด โดยการทำเป็น WINDOW DISPLAY ในภาคภาษาไทยแสดงผลได้ 40 คอลัมน์ 12 บรรทัด และ 80 คอลัมน์ 12 บรรทัด ในเครื่องเดียวกัน

● MULTIPLE TERMINAL

เมื่อพ่วงกับ HARD DISK และ UNIBUS TM ทำให้สามารถทำเป็น LOCAL AREA NETWORK คือสามารถต่อเป็น MULTI STATION ได้ และใช้เป็นระบบ MULTI-USER หรือ MULTI-OPERATING SYSTEM ได้

พิเศษ ! ซื้อ CHERRY III-EX วันนี้ แถมโปรแกรมภาษาไทยและอังกฤษรวม 12 รายการ

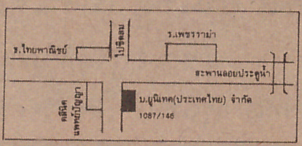
1. ไทยเวอร์ดโปรเซสซิ่ง (THAI WORD PROCESSING)
2. ระบบเวอร์ดชีทไทย (THAI SUPERCALC)
3. ระบบบริหารสินค้าคงคลัง (THAI INVENTORY CONTROL)
4. ระบบจ่ายเงินเดือน (THAI PAYROLL)
5. DBASE II ไทย (ระบบใหม่)
6. ระบบซื้อขายลดเช็คไทย
7. ระบบบัญชีรับ-จ่าย รายวันพร้อมใบส่งสินค้า
8. MBASIC ไทย (ระบบใหม่)
9. THAI APPLESOFT (THAI DOS 3.3)
10. 192 K CP/M
11. 192 K DOS 3.3
12. 192 K MULTIPLAN
13. 192 K PASCAL

การให้บริการ โปรดดูระบบ **RISK FREE INVESMENT PLAN** ของทางบริษัท สนใจติดต่อขอรายละเอียดได้ที่

บริษัท ยูนิเทค (ประเทศไทย) จำกัด

1087/146 ปากซอยจารุรัตน์ ถ.เพชรบุรี ตรงข้ามธนาคารไทยพาณิชย์

สาขาใหญ่ กรุงเทพฯ โทร. 251-6465



UNITECH (THAILAND) CO.LTD.

ระบบ RISK FREE INVESMENT PLAN สำหรับผู้ซื้อ CHERRY III

บริการการรับประกัน

การรับประกัน บริษัทฯ ให้การรับประกัน 1 ปี

บริการแก้ไข

ในกรณีเครื่องเสีย เมื่อได้รับการติดต่อจากผู้ใช้ จะมีเจ้าหน้าที่ของบริษัทฯ นำเครื่องที่ดีไปสับเปลี่ยน โดยใช้เวลารอคอยน้อยที่สุด เมื่อช่างทำการซ่อมแซมหรือแก้ไขเสร็จ จะนำเครื่องเดิมไปติดตั้งเพื่อใช้ต่อไป

บริการซ่อมบำรุง

การให้บริการทางด้านซ่อมบำรุง เมื่อพ้นกำหนดระยะเวลารับประกัน บริษัทฯ ให้บริการตรวจเช็คดูแลรักษาเครื่อง โดยทำสัญญากับบริษัทฯ เป็นรายปี คิดค่าบริการในอัตราที่ถูกกว่าเมื่อเปรียบเทียบกับบริษัทอื่น

บริการเกี่ยวกับภาษา

ภาษาที่ใช้ ผู้ซื้อสามารถเลือกภาษาใดภาษาหนึ่งเพิ่มเติมจากภาษาอังกฤษ คือ ภาษาไทย ภาษาญี่ปุ่น, หรือภาษาจีน โดยทางบริษัทฯ มิได้คิดเงินเพิ่ม

บริการระบบเงินผ่อน

บริษัทฯ จัดบริการเช่าซื้อให้กับผู้ที่ประสงค์จะซื้อในระบบเงินผ่อน

บริการรับซื้อคืน

บริษัทฯ รับ TRADE-IN ในราคายุติธรรม ในกรณีที่มีเครื่องคอมพิวเตอร์รุ่นใหม่ที่มีประสิทธิภาพ และคุณภาพดีที่ทางบริษัทฯ คัดเลือกมาเพื่อจำหน่ายและผู้ซื้อของบริษัทฯ มีความประสงค์จะซื้อ

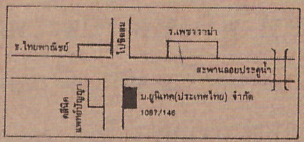
บริการด้านฝึกหัดอบรม

บริษัทฯ ได้จัดบริการด้านฝึกหัดอบรมการใช้เครื่อง การเขียนโปรแกรมคำสั่ง จัดดำเนินการทำโปรแกรมเกี่ยวกับงานด้านธุรกิจ การศึกษา และให้คำปรึกษาเรื่องระบบ และทางด้านเทคนิคอื่น ๆ

บริษัท ยูนิเทค (ประเทศไทย) จำกัด

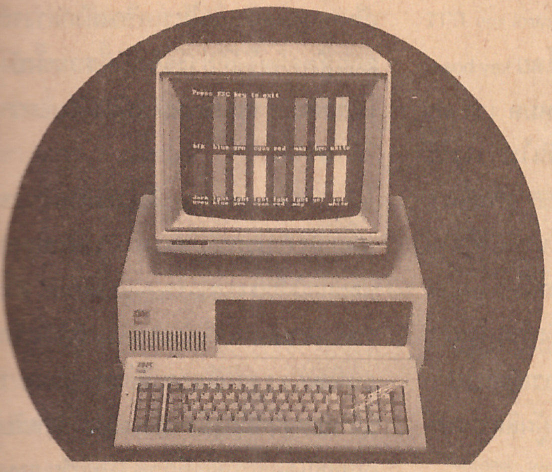
1087/146 ปากซอยจารุรัตน์ ถ.เพชรบุรี ตรงข้ามธนาคารไทยพาณิชย์

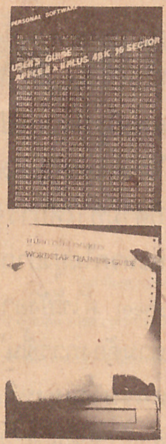
สาขาใหญ่ กรุงเทพฯ โทร. 251-6465



M·SOURCE

ผลิตภัณฑ์ใหม่ !





หนังสือวิสิแคลภาษาไทย

ราคาเล่มละ 70.-

คู่มือการใช้ Word Star ภาษาไทย

ราคาเล่มละ 140.-

IBM PC COMPATIBLE ที่สมบูรณ์แบบในคุณภาพและราคายุติธรรม รวมทั้งอุปกรณ์ต่าง ๆ สำหรับ 16 BIT เชิญแวะชมและสอบถามรายละเอียดได้ที่บริษัท ไมโครซอร์ส จำกัด

PC Compatible

PRICE LIST

1) Standard Unit (256k RAM, DRIVE 5¼" DS/DD 360k × 2, Printer Post		75,000 บาท
2) Monochrome Monitor		8,000 บาท
3) Monochrome card		8,000 บาท
4) RGB Color Graphic/Video Composit Card		10,000 บาท
5) RGB Color Monitor		18,000 บาท
6) Diskettes 5¼" DS/DD	1,500.-	1,800 บาท
7) Software For IBM PC	1,000.-	10,000 บาท

Software For IBM PC.

DOS 1.1, 2.0
Visitrend/Plot
Lotus 123
dBASE II 2.3, 2.4
MULTIPLAN
Diagnostic
Spooler Buffer
Word Star 3.2, 3.3
Spellstar
Compaq DEMO
IBM DEMO Software
PC TUTOL
Supercalc Ⅲ
The Norton Utilities
Game

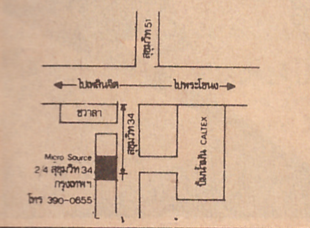
BOOK

GRIDE TO OPERATION	263
BASIC (FOR IBM PC)	336
DISK OPERATION SYSTEM	315
LOTUS 123 CUSTOMER ASSURANCE PLAN	441
YOUR IBM PC	315
IBM PC EXPANSION & SOFTWARE GUIDE	378
PERFECT FILER	189
IBM TECHNICAL REFERENCE MANUAL	336
UCSD PASCAL REFERENCE	315
FORTRAN - 77 REFERENCE	315
UPDATES FOR IBM PC COBOL COMPILER	336
BEGINNER'S GUIDE	263
ASSEMBLER REFERENCE	210
FORTRAN COMPILER	231
SUPERCALC 2 USER'S GUIDE & REFE MANUAL	210
HARDWARE NAINTENANCE AND SERVICE	420
IBM PC CONVERSION HANDBOOK OF BASIC	320
IBM DATA FILE (A BASIC TUTORIAL)	400
BUSINESS APPLICATION FOR IBM PC	450
BASIC FOR BUSINESS FOR IBM PC	287
100 READY-TO RUN PROGRAMS & SUBROUTINES FOR IBM PC	360
BEGINNER'S GUIDE	263
USCD PASCAL REFERENCE	315
ASSEMBLER REFERENCE FOR USED P-SYSTEM	210
IBM PERSONAL COMPUTER	320
THE NORTON UTILITIES	200
INSIDE THE IBM PC	360
DOCUMENTATION INSIDE THE IBM PC	220
HOW TO GET STARTED WITH MS-DOS	290
VISICALC FOR IBM PC	230
33 GAMES OF SKILL & GHANCE FOR IBM PC	320
BASIC FOR BEGINNERS	290
PC GRAPHICS	360
MICROSPELL USER'S MANUAL	150
POWER OF LOTUS 1-2-3	450
ALL ABOUT 1-2-3	200
CP/M 86 OPERATING SYSTEM COMMAND SUMMARY	180

ถ้าท่านต้องการ **IBM PC COMPATIBLE, APPLE COMPATIBLE** เรายินดีให้คำปรึกษาแก่ ท่านอย่างเป็นกันเอง

(บริการคือสิ่งที่เรามีให้แก่ท่าน)

สินค้าเหล่านี้ท่านสนใจติดต่อ บริษัท ไมโครซอร์ส จำกัด เราบริการท่านตั้งแต่ 9.00-20.00 น. จันทร์-เสาร์



บริษัท ไมโครซอร์ส จำกัด
2/4 สุขุมวิท ซอย 34 กรุงเทพฯ 10110
โทร. 3900655

MICRO SOURCE
CO.LTD 2/4 SUKUMVIT 34
BANGKOK 10110 TEL. 3900655

Framework ซอฟท์แวร์รวมจาก-Ashton-Tate.

นิวยอร์ค - บริษัท Ashton-Tate ผู้ผลิต dBASE II ที่เรารู้จักกันดีได้ออกซอฟท์แวร์ตัวใหม่ที่เป็นซอฟท์แวร์รวม (integrated Software) เพื่อแข่งขันกับ Lotus Development ซึ่งนอกจากจะมี 1-2-3 เป็นซอฟท์แวร์รวมยอดนิยมแล้วยังออกซอฟท์แวร์ตัวใหม่คือ Symphony มาอีกด้วย

ซอฟท์แวร์รวมของ Ashton-Tate นี้ชื่อว่า Framework มีการทำงานโดยแบ่งจะเป็นหลาย Window ประกอบด้วยฟังก์ชั่นทางด้าน Word Processing, file management, graphic และ Spreadsheet คาดว่า Framework นี้จะเป็นผลิตภัณฑ์หลักของ Ashton-Tate ในปี 1984 แต่ก็ยังมีคู่แข่งที่น่ากลัวคือ Symphony และ New Context MBA ของ Context Management System.

Framework นี้สามารถใช้ได้กับ IBM PC และ compatible ทั้งหลายโดยต้องการ RAM 256 K กับดิสค์ไดรฟ์ 2 ตัว โปรแกรมนี้สามารถแสดงภาพเป็นกราฟวงกลม กราฟแท่งและกราฟชนิดอื่น ๆ บนจอภาพ สีเดียว (monochrome) โดยไม่จำเป็นต้องใช้แผงวงจรเพิ่มเติมทางด้านกราฟฟิค ภาพหรือกราฟใด ๆ ที่สร้างขึ้นจะสามารถนำไปรวมกับข้อความใน Text ได้ เมื่อพิมพ์ออกมาพร้อมกันก็จะอำนวยความสะดวกได้อย่างมากในการจัดทำรายงานต่าง ๆ

ฟังก์ชั่นต่าง ๆ ของ Framework สามารถเลือกได้จาก menu, function key ที่โปรแกรมไว้หรือ control key การควบคุม cursor ทั้งหมดทำโดย key board (ไม่ใช้ mouse) คำสั่งมีทั้งหมด 130 คำ และมีรูปแบบอย่างเดียวกันไม่ว่าจะใช้ใน Spreadsheet, word processor หรืออื่น ๆ ก็ตามทำให้การจดจำและนำไปใช้งานสะดวกไม่ยุ่งยาก นอกจากนี้ใน Spreadsheet ผู้ใช้ยังอาจสร้างฟังก์ชั่นใหม่ ๆ ตามต้องการขึ้นได้จากการเขียนโปรแกรมขึ้นเองหรือซื้อจากผู้ผลิตซอฟท์แวร์รายอื่น ๆ

Osborne PC จะออกสู่ตลาดเร็ว ๆ นี้

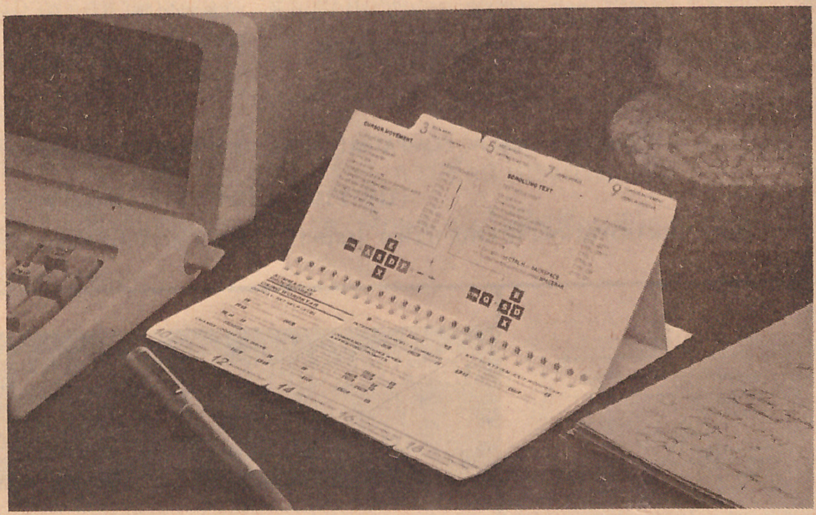
บริษัท Osborne Computer ซึ่งถูกศาลสั่งให้ล้มละลายเมื่อปีก่อนและกำลังดำเนินการภายใต้โครงการฟื้นฟูบริษัทใหม่ภายใต้การดูแลของศาล ได้นำเครื่อง Osborne PC ซึ่ง compatible กับ IBM PC ออกแสดงในงาน West Coast Computer Fair เมื่อเร็ว ๆ นี้ แต่ยังไม่แถลงเป็นทางการว่าเมื่อไหร่เครื่องดังกล่าวจะออกสู่ตลาดได้อย่างจริงจัง รวมทั้งยังไม่กำหนดราคาแน่นอน เพียงแต่บอกว่า "ต่ำกว่า $3000" เท่านั้น

Osborne PC สีน้ำเงินหนัก 28 ปอนด์ (เบากว่า IBM PPC เล็กน้อย) ประกอบด้วย 256 K. floppy disk drive 2 ตัว, 2 serial และ 1 parallel port, คีย์บอร์ดขนาดมาตรฐานและจอสีอำพันขนาด 7 นิ้ว นอกจากนี้ยังมี RGB Adapter สำหรับต่อมอนิเตอร์สี เพื่อที่สำหรับขยาย RAM ได้ถึง 512K และ socket สำหรับ 8087 mathematic coprocessor

บริษัท Osborne อ้างว่า Osborne PC นี้สามารถ RUN ซอฟท์แวร์ที่ใช้กับ PC-DOS และ MS-DOS ได้เช่นเดียวกับ IBM PC ทุกประการ

Quick Reference Guide ใกล้มือ

คุณเคยรู้สึกรำคาญไหมครับเวลาที่ใช้โปรแกรมสำเร็จรูปอันหนึ่งอยู่แล้ว



นึกคำสั่งไม่ออก จะใช้คำสั่ง Help ดู (ถ้าโปรแกรมนั้นมีให้ใช้) ก็ไม่ทันใจ ถ้าเป็นเช่นนั้น วิธีการนี้อาจจะเหมาะกับคุณมากกว่า

บริษัท Reston Computer Group ได้ออกสมุดคู่มือการใช้งานโปรแกรมสำเร็จรูปต่าง ๆ ขนาดเหมาะมือ และการจัดรูปแบบคำอธิบายก็สะดุดตาง่ายแก่การทำความเข้าใจ โดยมีคำอธิบายตั้งแต่ระดับพื้นฐาน สำหรับทำความเข้าใจฟังก์ชั่นใหม่ ๆ ที่คุณอาจไม่เคยใช้ หรือคำอธิบายแบบย่อสำหรับอ้างอิงถึง Command หรือ Control ต่าง ๆ สำหรับฟังก์ชั่นที่คุณเข้าใจการทำงานดีแล้วแต่จำคำสั่งไม่ได้ ลักษณะของสมุดคู่มือ MICROREF ก็อย่างที่เห็นในรูปนี้แหละครับ

MICROREF มีทั้งสำหรับ WORDSTAR, MULTIPLAN, SUPERCALC, VisiCalc, dBASEII และ LOTUS 1-2-3 สนใจติดต่อได้ที่ Reston Computer Group, A Prentice-Hall Company, 11480 Sunset Mills Rd., Reston, VA22090

ซอฟท์แวร์รวม (Integrated Software) สำหรับ Apple II

ยุคนี้เป็นยุคของการทำงานหลายอย่างพร้อม ๆ กัน ตั้งแต่ระบบ Integrated Software ของ Lisa และ Macintosh ตามมาด้วยการแข่งขันกันอย่างดุเดือดในตลาดของ Integrated Software ระหว่าง Lotus 1-2-3, Supercalc[3] และอื่น ๆ จนถึงยุคที่ Window กลายเป็นความนิยมอย่างใหม่ที่ผู้ผลิตซอฟท์แวร์รายใหญ่ ๆ พากันส่งเสริมด้วยการออกผลิตภัณฑ์ใหม่ออกมาอย่างไม่หยุดยั้ง มีทั้งสำหรับ IBM PC และเครื่องอื่น ๆ ทั้งที่ใช้ CPU เบอร์ 8086/8088 และ 68000

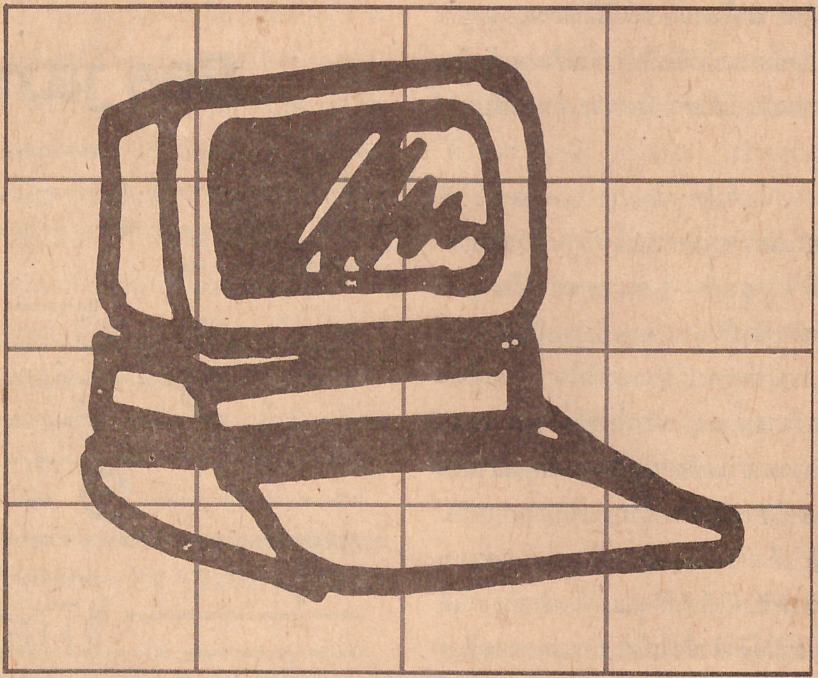
เท่าที่ผ่านมาผู้ที่มีแต่เครื่อง 8 บิท เช่น Apple II ก็ได้แต่นั่งมองตาปริบ ๆ เพราะไม่มีใครผลิตซอฟท์แวร์รวมสำหรับเครื่องเล็กขนาดนี้ออกมาเลย แต่เดี๋ยวนี้บริษัท Arktronics ได้นำเทคโนโลยีของซอฟท์แวร์รวมมาสู่เครื่อง 8 บิท แล้วครับ ซอฟท์แวร์รวมตัวนี้มีชื่อว่า Jane ใช้ได้กับเครื่อง Apple II, II Plus, II e. Commondore 64, และที่กำลังจะออกมาเพิ่มอีกก็คือสำหรับ Atari และ IBM PC (รวมทั้งที่ Compatible ด้วย)

Jane ประกอบด้วย Janewrite ซึ่งเป็น Word Processor, Janecalc ซึ่งเป็น

ข่าวคอมพิวเตอร์

Electronic Spreadsheet และ Janelist ซึ่งเป็น File Manager ลักษณะการใช้งานจะเป็น Window และมีรูปแทนฟังก์ชั่นการทำงานต่าง ๆ ซึ่งเลือกใช้งานได้จาก Mouse ถึงแม้ว่าฟังก์ชั่นการทำงานของโปรแกรมต่าง ๆ ของ Jane จะไม่ซับซ้อนหรือมีประสิทธิภาพสูงเท่ากับโปรแกรม Word Processor, Spreadsheet หรือ File Manager โดยเฉพาะก็ตาม

Jane มีราคาในอเมริกา 295 เหรียญ โดยบริษัท Arktronics 113 South Fourth Avenue, Ann Arbor, Michingan 48104



Context MBA ฉบับปรังปรุงใหม่

ในยุคที่ Lotus 1-2-3 ออกมาใหม่ ๆ นั้น คู่แข่งที่ขับเคี่ยวกันอย่างดุเดือดก็คือ Context MBA ซึ่งว่ากันว่าเป็น Integrated-Software ที่มีประสิทธิภาพไม่น้อย เสียแต่การใช้งานไม่สะดวกเท่ากับ Lotus 1-2-3 แต่เดี๋ยวนี้บริษัท Context Management System ได้ปรับปรุงผลิตภัณฑ์ออกมาใหม่เรียกว่าเป็น The New Context MBA และ Corporate MBA ซึ่งรวมเอา Spreadsheet, Infromation, Management, Business Gaphic, Executive Word Processing และ Teleccommunication Function เข้าไว้ใน Package เดียว นอกจากนี้ยังมี Context 500 เป็น After Sale Support อีกด้วย

Context Management System Inc. อ้างว่าได้ปรับปรุงซอฟท์แวร์ตัวใหม่นี้ในหลาย ๆ ด้าน แต่จะสามารถแย่งความนิยมคืนมาจาก Lotus 1-2-3 ที่นำหน้าไปหลายเดือนแล้วได้หรือไม่นั้นก็คงต้องคอยดูกันต่อไป สนใจติดต่อได้ที่ Context Management System Inc. 23868-Hawthrone Blvd. Torrance, CA90505

ความลับของ IBM

รายงานข่าวที่ยังไม่เป็นทางการแจ้งว่า IBM กำลังพัฒนาคอมพิวเตอร์สำหรับผู้ใช้หลายคน (Multiuser) โดยใช้ไมโครโปรเซสเซอร์เบอร์ 80186 (ซึ่งเป็นไมโครโปรเซสเตอร์ 16 บิทจริง) ของอินเทล ทั้งนี้หลังจากที่ระงับแผนการที่จะใช้ 80286 ซึ่งก้าวหน้ากว่าเนื่องจาก 80286 ประสบปัญหาในการผลิต

นอกจากนี้ยังมีรายงานว่า IBM กำลังพัฒนา Operating Environment (โปรแกรมที่จัดภาวะแวดล้อมในการทำงานของโปรแกรมอื่น ๆ เสียใหม่ เป็นตัวกลางระหว่างโปรแกรม Application กับ Operating System เช่นจัดระบบ Window โดยรับข้อมูลจาก Application ต่าง ๆ จัดรูปแบบใหม่และควบคุมการทำงานจริง ๆ ผ่านทาง Operating System) ที่มีชื่อรหัสว่า "Glass" โปรแกรมนี้มีเค้าว่าจะเป็นคู่แข่งที่สำคัญของ VisiOn ของ VisiCorp และ Window ของ Microsoft ทีเดียว

ข่าวแจ้งว่า Glass จะใช้ Mouse และมีระบบ Window เช่นเดียวกับโปรแกรมของ VisiCorp และ Microsoft แต่ยังไม่แน่ว่า Operating Environment นี้จะ Run ภายใต้PC-DOS หรือ Operating System ตัวใหม่ที่ IBM จะพัฒนาขึ้นกันแน่ สรุปแล้วข่าวนี้ยังไม่มีอะไรแน่นอนสักอย่าง ความลับของ IBM ยังเป็นเรื่องที่ต้องเดากันต่อไป

ข่าวจาก Digital Research

Concurrent CP/M-86 ของ Digital Research ต้นตำรับ CP/M เป็น Operating System ที่สามารถทำงานได้ถึง 4 อย่างพร้อม ๆ กัน ส่วน Concurrent DOS ก็คือ Concurrent CP/M-86 version 3.1 ที่บริษัท Digital Research ได้พัฒนาขึ้นใหม่โดยรวมเอาความสามารถทางด้าน Window, network และ multiuser เข้าไปด้วย นอกจากนี้ยังมี "PC-mode" ที่ทำให้สามารถนำโปรแกรมใช้งานต่าง ๆ ที่ปกติ Run-ภายใต้ PC-DOS ของ IBM PC มาใช้ Run ภายใต้ Concurrent DOS ได้อีกด้วย

ขณะนี้ Digital Research ได้ประกาศจับมือกับ Motorola ผู้ผลิตไมโครโปรเซสเซอร์ 68000 ที่ใช้ใน Macintosh พัฒนา Version ใหม่ของ Concurrent Dos สำหรับใช้กับ 68000.นับว่าเป็นการขับเคี่ยวกับ Microsoft คู่ปรับเก่าที่กวาดตลาด 16 บิท 8086/8088 ไปครองด้วย MS-DOS และ PC-DOS จน CP/M-86 ของ Digital Research แทบตกขอบไปเลย

ข่าวจาก Digital Research อีกอันหนึ่งคือ Gold Card ซึ่งเป็น CP/M Card สำหรับ Apple II, II Plus และ IIe ที่ Digital Research ได้พัฒนาขึ้นใหม่โดยใช้ไมโครโปรเซสเซอร์ Z-80 B และมี Chache Disk (คล้าย ๆ กับ RAM Disk คือเก็บข้อมูลบางส่วนจากดิสค์ไว้ใน RAM เลย) ทำให้เพิ่มความเร็วในการทำงานขึ้นอีก โปรแกรมควบคุมระบบที่ใช้ได้คือ CP/M Version พิเศษ (CP/M Plus) และยังให้ Output ทางจอภาพเป็น 80 คอลัมน์ในตัวอีกด้วย น่าเสียดายที่เพิ่งจะมาออกเอาตอนนี้

เร่งสปีด Apple II

บริษัท Titan Technologies Inc. ได้ออก Card Accelerator IIe ซึ่งมี CPU 6502 รุ่นความเร็วสูง, RAM high speed 64 K และ Language Card ที่ built-in ทั้งหมดนี้จะทำให้โปรแกรมต่าง ๆ ที่ใช้ CPU 6502 ใน Apple II, II Plus หรือ IIe ทำงานเร็วขึ้น $3\frac{1}{2}$ เท่าทั้งนี้ตามคำอ้างของผู้ผลิต

ในการใช้งาน Accelerator IIe Card นี้จะ Transparent หรือโปร่งใส คือโปรแกรมต่าง ๆ จะทำงานได้ตามปกติเสมือนไม่มี Card นี้อยู่ เพียงแต่เร็วขึ้นเท่านั้น และ Card นี้ยังสามารถเลือกใช้หรือไม่ใช้ก็ได้โดยการควบคุมจากคีย์บอร์ด

บริษัท Titan Technologies ยังได้ออก Card Accelerator ทำนองเดียวกันนี้สำหรับ IBM PC อีกด้วย ซึ่งจะทำให้ IBM PC ทำ งานเร็วขึ้นอีกหลายเท่า สนใจติดต่อบริษัท Titan Technologies Inc. ,310 West Ann Street, Ann Arbor, MI 48104

Bubble Memory สำหรับ IBM PC

Helix Laboratories, Inc. ได้ผลิต Bubble Momery Board ความจุครึ่งเมกกาไบท์ ในชื่อ HELIX PC BUBBLE DISK สำหรับ IBM PC หรือ XT ซึ่งมีคุณสมบัติดังนี้

- Non-Volatile ข้อมูลยังอยู่เมื่อปิดไฟ
- เร็วกว่า Floppy disk ประมาณแปดเท่า
- มีค่า Mean Time Between Failure (MTBF) หรืออายุการใช้งานเฉลี่ยระหว่างการเกิดความผิดพลาดแต่ละครั้งมากกว่า 20 ปี

สนใจติดต่อ Helix Laboratories, Inc. Marketting Division-Suite 720 11601 Wilshire Blud, Los Angeles, CA 90025

ฮิตาชิเปิดตัวเครื่องที่มีแอดเดรส 31 บิท

โตเกียว ฮิตาชิเผยโฉมเครื่องระดับเมนเฟรมที่ใช้แอดเดรสซิ่งโหมด 31 บิท ซึ่งสามารถขยายไปเป็น Virtual Memory ได้ถึง 2 G bytes ระบบเครื่อง Compatible กับเมนเฟรมของ IBM และทำงานด้วยโปรเซสเซอร์ของฮิตาชิชื่อ M-280 และ M-260 ภายใต้ Operating System VOS/ES1 ทั้ง ไอ บี เอ็ม และฟูจิตสึได้แนะนำเครื่อง 31 บิท ไปเรียบร้อยแล้วก่อนหน้าฮิตาชิจะเข้ามายังตลาด 31 บิทนี้

ถ้าจะถามว่าทำไมฮิตาชิรีบเร่งเข้าสู่ตลาดนัก คำตอบก็คือว่าฮิตาชิเชื่อว่าการเพิ่มปริมาณของผู้ใช้โปรแกรมประยุกต์และการพัฒนาโปรแกรมที่ทวีความซับซ้อนขึ้นไปอีกทำให้จำเป็นต้องมีหน่วยความจำเกินกว่า 16 เมกกาไบท์

ระบบฮิตาชิ 31 บิท สามารถสวิทช์เลือกระหว่างแอดเดรสซิ่งโหมด 24 บิท และ 31 บิทได้นอกเหนือจากนั้น Operating System ที่เกินกว่า 16 เมกกาไบท์ สามารถ Relocate ในหน่วยความจำที่อ้างถึงโดยแอดเดรส 31 บิทนี้ได้ ด้วยเหตุนี้ จึงเพิ่มเนื้อที่สำหรับระบบไฟล์

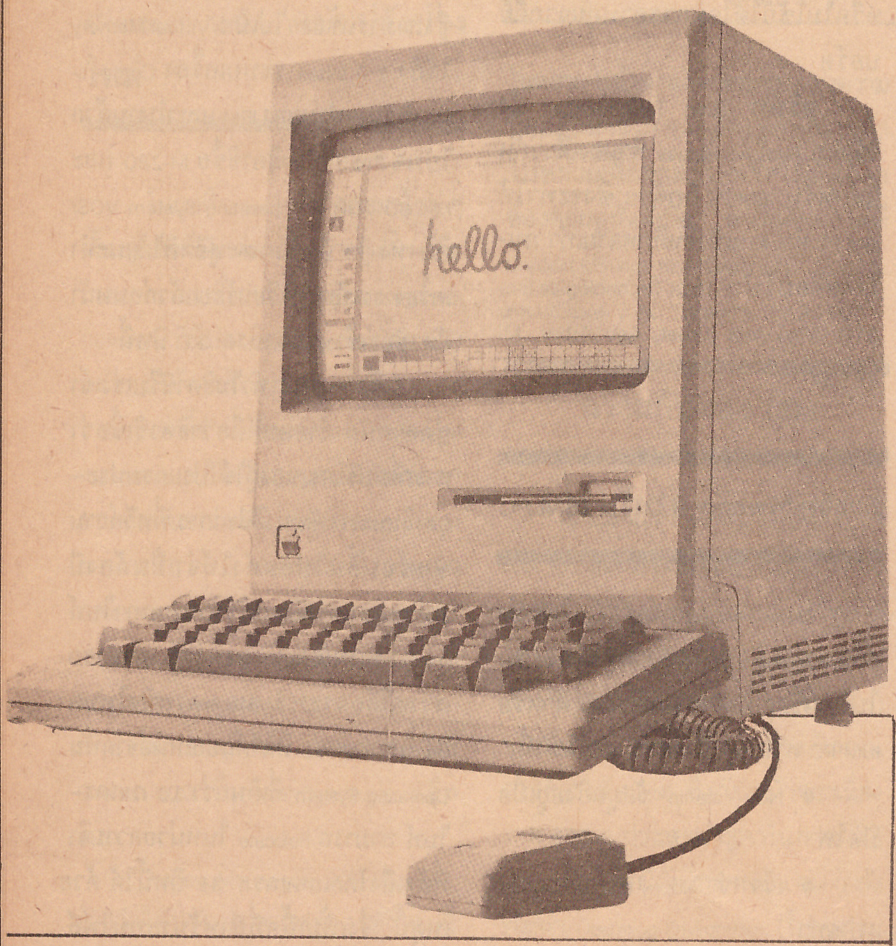
และจำนวนเทอร์มินอลที่สามารถติดตั้งได้ตามไปด้วย

ในการปรับปรุงระบบคอมพิวเตอร์ที่มีอยู่แล้วให้ใช้โปรเซสเซอร์ M-280 และ M-260 ได้จะต้องเสียค่าใช้จ่ายประมาณ US$ 3100 ส่วน Operating System ที่ปรับปรุงขึ้นใหม่ คือ VOS 3/ES1 ซึ่งพัฒนาขึ้นจาก VOS 3 โดยส่วนที่เกินกว่า 16 เมกกาไบท์จะ Relocate ใน Virtual Memory ตามปกติ Operating System จะถูกเก็บใน Main Memory อยู่ตลอดเวลาโดยไม่มีการ Relocate

VOS 3/ES1 มีอยู่ 2 รุ่นคือ VOS 3/ES1-JSS3 ซึ่งใช้กับการทำงานของโปรเซสเซอร์ Single Processor ตัวเดียวและอีกรุ่น VOS 3/ES1-JSS4 สนับสนุนระบบ-Multiprocessor

Operating System ฮิตาชิ VOS 3/ES นี้เทียบได้กับ MVS/SP Version 2 ของ IBM และ OSIV/F4 MSP ของ Fujitsu แต่อย่างไรก็ตามเรื่องสิทธิของฮิตาชิใน Operating System ชุดของ VOS 3 นี้ยังเป็นปัญหาอยู่ทั้งนี้เนื่องจากกรณีที่ฮิตาชิถูก IBM ฟ้องข้อหาจารกรรมทางเทคโนโลยี่เมื่อเร็ว ๆ นี้

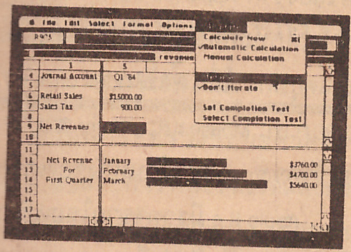
มาสแรกของปี 1984



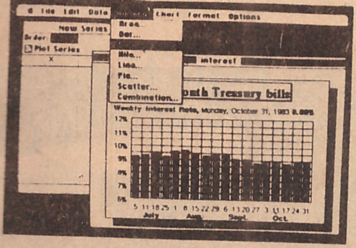
ไมโครซอฟท์ออกโปรแกรมสำหรับ Macintosh

ไมโครซอฟท์เจ้าพ่อซอฟท์แวร์ได้ประกาศตัวโปรแกรมของตนสำหรับใช้ใน Apple Macintosh แล้วเมื่อเร็ว ๆ นี้ 5 ตัวด้วยกันคือ Mutiplan, Word, Chart, File และ Microsoft BASIC โปรแกรมของไมโครซอฟท์ทุกตัวจะใช้จุดเด่นทางด้านกราฟิคของ Mac รวมทั้งความสะดวกสบายของการใช้ Mouse ในการเลือกฟังก์ชั่นหรือคำสั่งต่าง ๆ นอกจากนั้นยังสามารถใช้ในการติดและเชื่อมข้อมูลต่าง ๆ ใน window ของ Mac และในระหว่างโปรแกรมของไมโครซอฟท์เองด้วย

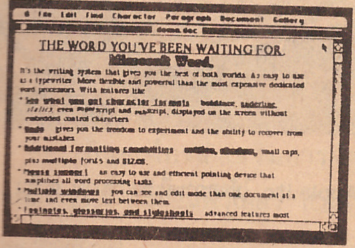
Multiplan ของ Macintosh มีการเพิ่มคำสั่ง Undo ซึ่งทำให้เครื่องสามารถแก้สิ่งที่เปลี่ยนแปลงไปครั้งสุดท้ายให้กลับคืนมาเหมือนเดิม ส่วน Word นั้นสามารถใช้จุดเด่นของ Mac ด้านกราฟฟิคได้อย่างเต็มที่ และมีแบบของตัวอักษรได้ตามแบบของ Mac โดยแสดงลักษณะอักษรได้ทั้งตัวธรรมดาและตัวพิเศษต่าง ๆ เหมือนกับที่จะพิมพ์ออกจริงทาง printer โปรแกรม Chart นั้นผู้ใช้สามารถ edit และจัด format ข้อมูลต่าง ๆ ได้บนจอโดยตรง และยังรับข้อมูลจาก Multiplan และ File ได้อีกด้วย ส่วนโปรแกรม File นั้น สามารถจัดลักษณะ data entry ได้หลายแบบ และ Microsoft BASIC ก็ใช้จุดเด่นของไมโครโปรเซสเซอร์ 68.000



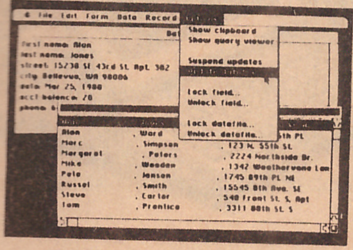
Microsoft Multiplan



Microsoft Chart



Microsoft Word



Microsoft File

โดยสามารถใช้ตัวเลขได้ถึง 14 หลัก ใช้ String มีความยาวได้ถึง 32,767 ตัวอักษร และมี Window ได้ 3 ช่อง

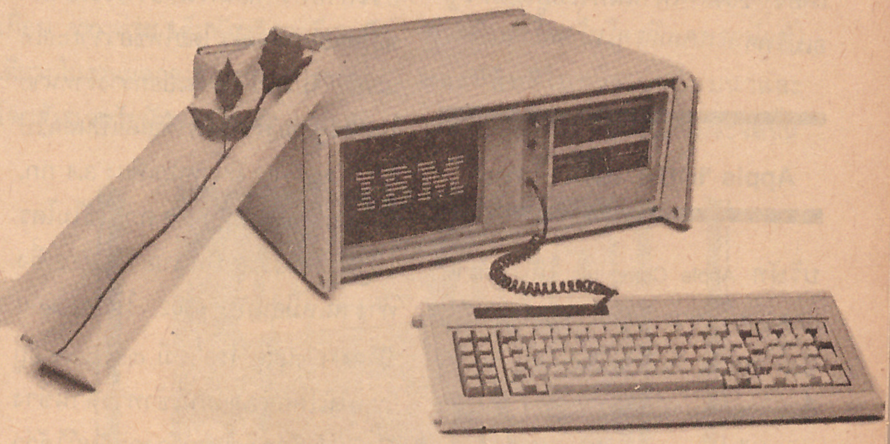
Microsoft Multiplan,Word และ File ในสหรัฐ ราคา $195 ส่วน Chart ราคา-125$ และ BASIC ราคา $150

ของใหม่จาก IBM

IBM ได้แนะนำผลิตภัณฑ์ใหม่ของตนออกมาพร้อม ๆ กันหลายชนิดด้วยกัน ตัวแรกได้แก่ IBM PC แบบกระเป๋าหิ้ว หรือเรียกชื่อย่อว่า PPC (Portable Personal Computer) ซึ่งมีหน่วยความจำ-256 K RAM และ Disk Drive 1 ตัว ในราคา $2795 disk drive ตัวที่สองจะต้องเพิ่มเงินอีก $425 PPC มี expansion 5 slot ด้วยกันสำหรับต่อ Printer และ Communication Port จอภาพเป็นจอสีเหลืองอำพัน (Amber) ขนาด 9 นิ้ว และภายในมี Interface สำหรับจอสีด้วย PPC มีขนาด 20×17×8 นิ้ว หนัก 30 ปอนด์ พร้อมด้วยกระเป๋าหิ้ว PPC สามารถ run software ที่ใช้กันสำหรับ IBM PC ได้ส่วนใหญ่ ยกเว้นโปรแกรมที่เขียนขึ้นให้ใช้ภายใต้ MS-DOS 1.1 หรือที่ต้องการ Monitor ขาวดำต่างหาก

นอกจากนั้น IBM ยังได้ประกาศผลิตภัณฑ์ใหม่ซึ่งสามารถเชื่อม IBM PC, PPC, PC XT และ PCjr เข้ากันได้เป็นจำนวนมากที่สุดถึง 64 ตัว โดยแต่ละตัวต้องการ Cluster program ราคา $92 ส่วน Cluster Adapter สำหรับ PC, PPC และ PC XT ราคา $375 ส่วนของ PCjr นั้นราคา $400

สายเคเบิลหลัก ซึ่งมีความยาวได้ถึง 3,280 ฟุต HARD DISK ใน PC XT หรือใน Expansion Unit สามารถใช้เป็น Master ของ cluster ได้ Cable kit อันประกอบด้วยสายเคเบิล 9 ฟุต 2 เส้น และ Main Cable ยาว 32 ฟุต ราคาชุดละ $110 แต่ผู้ใช้สามารถใช้สายเคเบิลขนาด 75 โอห์มมาใช้แทนก็ได้



ออสบอร์นจะฟื้นคืนชีพ

ข่าวจากคาลิฟอร์เนียแจ้งว่า บริษัท Osborne Computer กำลังจะพ้นจากสภาพล้มละลาย และจะสามารถดำเนินการต่อไปได้ตามปกติประมาณกลางปีนี้ ทั้งนี้เป็นคำแถลงของนางไซด์ แมคเรย์โนล์ด รองประธานฝ่ายปฏิบัติการของบริษัท

บริษัท Osborne Computer ต้องคำพิพากษาให้อยู่ในสภาพล้มละลายตั้งแต่เดือนกันยายน ปีที่แล้ว ตั้งแต่นั้นมาฝ่ายบริหารจำนวน 4 คนของบริษัทได้ประสานงานอย่างใกล้ชิดกับคณะกรรมการเจ้าหนี้จำนวน 17 คน ในการจัดโครงสร้างบริษัทใหม่ เพื่อให้สามารถดำเนินการต่อไปได้ ขณะนี้แผนการดังกล่าวได้รับการอนุมัติโดยศาลแล้ว

นางแมคเรย์โนล์ด ได้กล่าวต่อไปว่า นโยบายของบริษัทจะเน้นหนักไปทางด้านต่างประเทศมากกว่าในประเทศ (สหรัฐอเมริกา) กิจกรรมของบริษัทจะอยู่นอกประเทศประมาณ 80% และมีเพียง 20% เท่านั้นที่จะอยู่ในสหรัฐอเมริกา

Apple ขายดี แต่กำไรหด

บริษัท Apple Computer ได้ประกาศยอดขายในช่วง ไตรมาสที่ 2 ของปีนี้ ซึ่งสิ้นสุด ณ วันที่ 30 มีนาคม ยอดขายที่แถลงออกมามีมูลค่า $300.1 ล้านซึ่งสูงกว่าไตรมาสเดียวกันของปีที่แล้วถึง 32% แต่ยอดกำไรมีค่าเพียง $9.1 ล้าน ต่างจากปีที่แล้วซึ่งมีมูลค่า $23.9 ล้าน ทั้งนี้เพราะ Apple มีค่าใช้จ่ายด้านการพัฒนาผลิตภัณฑ์สูงมากในระยะนี้ ผลิตภัณฑ์ที่เป็นยอดขายหลักในขณะนี้คือ Apple II e ในขณะเดียวกัน Macintosh ซึ่งออกมาใหม่ในระยะนี้ก็ขายดีจนผลิตไม่ทัน ถึงแม้จะสร้างโรงงานใหม่เพิ่มขึ้นที่ฟรีมอนท์ คาลิฟอร์เนียแล้วก็ตาม การที่ Apple มีกำไรลดลงใน 2 ไตรมาสแรกนี้ ไม่ได้เป็นที่หนักใจแก่ฝ่ายบริหารของ Apple แต่อย่างใด เพราะได้คาดการณ์และประกาศไว้ล่วงหน้าตั้งแต่ปลายปีที่แล้ว อย่างไรก็ตามบริษัท Apple ก็ยังเป็นบริษัทที่มีฐานะทางการเงินมั่นคงมากบริษัทหนึ่ง โดยมีเงินสดในมือถึง $203.4 ล้าน และเงินลงทุนระยะสั้น และหุ้นในตลาดอีก $401.5 ล้าน เมื่อสิ้นสุดไตรมาส

Apple II c กระเป๋าหิ้วจาก Apple

บริษัท Apple Computer Inc ได้ประกาศตัว คอมพิวเตอร์กระเป๋าหิ้วของตนออกมาแล้ว โดยให้ชื่อว่า II c (II c นะครับ ไม่ใช่ II e) ซึ่งจะออกวางจำหน่ายในสหรัฐอเมริกาตั้งแต่เดือนนี้เป็นต้นไป

Apple II c มีน้ำหนักเพียง 3.4 กก. น้ำหนัก Battery หนัก 2.2 กก. ใช้ไมโครโปรเซสเซอร์ 8 บิท เบอร์ 65C02 ซึ่งทำงานเหมือนกับ 6502 ใน Apple II มี RAM ขนาด 128 K มี disk drive อยู่ภายในโดยสอดแผ่นเข้าทางด้านข้าง มีสวิทช์เลือกความกว้างคอลัมน์ได้ว่าเป็น 40 หรือ 80 คอลัมน์ คีย์บอร์ดเป็นแบบ 62 คีย์ ซึ่งเป็นแบบเดียวกับ Apple II e เนื่องจาก Apple II c ออกแบบมาให้มีขนาดเล็กกะทัดรัด จึงไม่มี expansion slot แต่อุปกรณ์อื่น ๆ เช่น disk drive ตัวที่ 2, จอภาพ, Mouse และ printer สามารถต่อเข้าทาง port ทางด้านหลังของเครื่อง

Apple II c มีราคาขายในอเมริกาเพียง $1295 ซึ่งนับว่ามีขนาดและราคาใกล้เคียงกับ PCjr ซึ่งเป็น home computer ของ IBM ที่ใช้ CPU 8/16 บิท 8088 มาก



Apple II c ใช้จอภาพสีเขียวขนาด 9 นิ้ว



Apple II c ใช้จอ LCD ขนาด 560 × 192 pixel

กระทรวงวิทยาศาสตร์ฯ จะตั้งบริษัทพัฒนาอุตสาหกรรมซอฟท์แวร์

ตามที่กระทรวงวิทยาศาสตร์ เทคโนโลยีและการพลังงานได้แต่งตั้งคณะกรรมการขึ้นคณะหนึ่ง เรียกว่า คณะกรรมการพัฒนาอุตสาหกรรมคำสั่งคอมพิวเตอร์ตามมติคณะรัฐมนตรีที่ให้ความสนับสนุนอุตสาหกรรมที่ใช้เทคโนโลยีชั้นสูงแล้วนั้น คณะกรรมการดังกล่าวได้ประชุมกันแล้วหลายครั้ง และได้มีมติให้ทำการจัดตั้งหน่วยงานซึ่งมีลักษณะเป็นบริษัทที่ไม่มุ่งผลกำไรขึ้นมาแห่งหนึ่ง เพื่อดำเนินงานต่าง ๆ ตามนโยบายที่ได้รับมอบหมายโดยจะเชิญบริษัทเอกชนและหน่วยงานต่าง ๆ ที่ทำธุรกิจด้านซอฟท์แวร์มาเป็นผู้ถือหุ้นโดยทางรัฐบาลจะให้ความสนับสนุนด้วยอีกทางหนึ่ง

วัตถุประสงค์ของการจัดตั้งบริษัทดังกล่าวมีดังนี้คือ

1. เป็นองค์กรส่งเสริมการจัดจำหน่ายชุดคำสั่งคอมพิวเตอร์ (Computer Software)

2. เป็นตัวแทนให้แก่สมาชิกในการติดต่อเจรจาจัดหาโครงการพัฒนาชุดคำสั่งคอมพิวเตอร์ทั้งในและนอกราชอาณาจักร

3. เป็นตัวแทนให้แก่สมาชิกในการจัดหางาน บริการ ผู้เชี่ยวชาญด้านคอมพิวเตอร์ทั้งในและนอกราชอาณาจักร

4. ให้บริการแก่สมาชิกในการจัดบริการคลังข้อมูลด้านคอมพิวเตอร์

5. เป็นองค์กรส่งเสริมด้านวิชาการคำสั่งคอมพิวเตอร์

6. เป็นตัวแทนจัดหาและจัดทำบริการด้านอื่น ๆ ที่เกี่ยวข้องกับธุรกิจชุดคำสั่งคอมพิวเตอร์

บริษัทแห่งนี้จะจดทะเบียนด้วยเงินลงทุนประมาณ 2 ล้านบาท

ขณะนี้คณะกรรมการดังกล่าวกำลังติดต่อกับบริษัทซอฟท์แวร์เฮ้าส์ต่าง ๆ เพื่อจัดประชุมพิจารณาความเป็นไปได้ของโครงการนี้อีกครั้งหนึ่ง

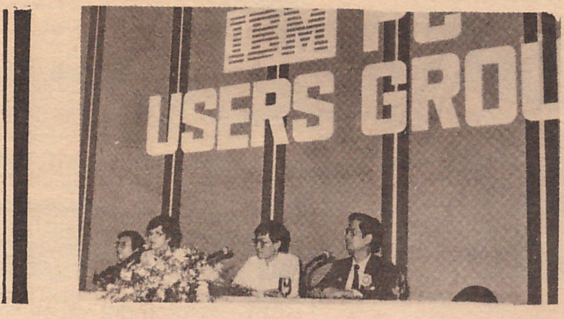
IBM PC User Club กำเนิดแล้ว

เมื่อวันที่ 24 พฤษภาคมศกนี้ที่โรงแรม Hyatt Central Plaza ห้องวิภาวดี พณฯ ท่านรองนายกรัฐมนตรี พิชัย รัตตกุล ได้ให้เกียรติมาเป็นประธานในการเปิดสัมมนาและสาธิตการใช้เครื่องโดยชมรมผู้ใช้เครื่องคอมพิวเตอร์ IBM PC

ในงานดังกล่าวนี้ได้เปิดโอกาสให้ผู้ที่ใช้เครื่อง IBM PC และผู้ที่สนใจสมัครเป็นสมาชิกของชมรม ปรากฏว่ามีผู้สนใจไปร่วมงานดังกล่าวเกือบ 800 ท่าน เต็มห้องวิภาวดีเลยทีเดียว ในงานนอกจากจะมีการบรรยายโดยผู้บรรยายผู้ทรงคุณวุฒิ อาทิเช่น ศาสตราจารย์ ดร.ศรีศักดิ์ จามรมาน และ อ.จ.ยืน ภู่วรวรรณ แล้ว ยังได้มีการจัดแสดงผลงานด้านโปรแกรมต่าง ๆ ของ IBM PC โดยบริษัทต่าง ๆ ที่ใช้เครื่องอยู่กว่า 10 บริษัท ซึ่งได้รับความสนใจจาก พณฯ ท่านรองนายก และผู้ร่วมชมงานเป็นจำนวนมากทีเดียว งานนี้ได้รับความร่วมมือเป็นอย่างดีจากบริษัทคอมพิวเตอร์ยูเนี่ยน ซึ่งเป็นตัวแทนจำหน่าย IBM PC ในประเทศไทย

จากการแถลงข่าวของ นายวิวัฒน์ ทองปุสสะ ประธานชมรม (ชั่วคราว) ได้แจ้งว่าทางชมรมจะได้ทำเรื่องขอจดทะเบียนเป็นสมาคมและจะทำการเลือกตั้งทั่วไป เพื่อเลือกประธานและกรรมการบริหารในเร็ว ๆ นี้

ท่านที่สนใจจะสมัครเป็นสมาชิกของชมรมนี้สามารถติดต่อได้ที่ที่ทำการชมรมชั่วคราวบริษัท คอมพิวเตอร์ยูเนี่ยน จำกัด โทร. 3115111-9, 311-6891 อัตราสมาชิก 300 บาท ต่อครึ่งปี และ 500 บาทต่อ 1 ปี โดยสมาชิกจะได้รับ Newsletter *(ได้รับความร่วมมือในการจัดทำโดยไมโครอิเลคทรอนิคส์)* และมีสิทธิในการเข้าสัมมนาและอภิปรายต่าง ๆ ที่ทางชมรมจะจัดขึ้น ■





เรื่องจากปก IBM PCjr : ดาวรุ่งดวงใหม่หรือลูกเป็ดขี้เหร่

ในที่สุด IBM ก็ได้ประกาศตัวไมโครคอมพิวเตอร์รุ่นเล็กของตนออกมา เป็นการยุติข่าวลือต่าง ๆ ที่วงการคอมพิวเตอร์ต่างกล่าวขวัญกันมาตั้งแต่ปลายปีที่แล้ว การปรากฏตัวของ-IBM PCjr ได้ก่อให้เกิดการวิพากษ์วิจารณ์กันอีกครั้งหนึ่ง โดยมีทั้งด้านบวกและด้านลบ

ในฉบับนี้เราจะขอแนะนำให้ท่านได้รู้จักกับ PCjr น้องใหม่ของ IBM นี้กันใน 2 แง่ด้วยกันคือ ในแง่ที่เปรียบเทียบกับเครื่องรุ่นพี่ คือ IBM PC และในแง่ของคุณสมบัติอื่น ๆ ที่แตกต่างกัน โดยเฉพาะทางด้านกราฟิคและด้านเสียง

ฮาร์แวร์ของ PCjr

IBM PCjr ใช้ไมโครโพรเซสเซอร์เบอร์เดียวกันกับ IBM PC นั่นคือ Inter 8088 ทำงานที่ความถี่ 4.77 MHz เช่นเดียวกัน มีหน่วยความจำเริ่มต้นที่ 64 K ขยายได้ถึง 128 K มีทั้งแบบ Version ราคาถูกที่ใช้เทปคาสเซทและแบบที่มี disk drive 1 ตัวติดมาด้วย เป็นแบบขนาดความจุ 360 K นอกจากนั้นยังมีอุปกรณ์ต่าง ๆ ที่สามารถเลือกมาประกอบ

กับระบบได้อาทิเช่น thermal printer, joystick และ modem card

IBM PCjr มีขนาด 13.9 x 11.4 x 3.8 นิ้ว และหนักเพียง 9 ปอนด์ ต้องมี transformer Adapter ต่างหากอยู่ข้างนอก สำหรับจ่ายกำลังไฟขนาด 33 วัตต์ ให้กับเครื่อง ด้านหน้าของเครื่องมีช่องเสียบคาร์ทริดจ์ ROM 2 ช่อง สำหรับเสียบโปรแกรมสำเร็จรูปหรือโปรแกรมเกมส์ต่าง ๆ

PCjr มี ROM ขนาด 64 K ซึ่งภายในบรรจุ power-on self test routine, cassette BASIC interpreter และ Operating System, I/O driver ตลอดจน disk bootstrap loader

จุดเด่นของ PCjr เห็นจะอยู่ตรงคีย์บอร์ด ซึ่งเป็นคีย์บอร์ดซึ่งไม่ใช้สาย โดยใช้แสงอินฟาเรดเป็นสื่อในการติดต่อกับ system unit แสงอินฟราเรดนี้สามารถจะใช้ในการติดต่อได้ไกลจากเครื่องถึง 20 ฟุต ตราบเท่าที่ตัวคีย์บอร์ดยังส่องไปยังตัวเครื่องได้ตรง ที่จริงเทคโนโลยีอันนี้ใช้กันมานานหลายปีแล้ว ในระบบควบคุมโทรทัศน์ แต่เพิ่งจะนำมาใช้กับไมโครคอม-พิวเตอร์เป็นครั้งแรก ทั้งนี้เพราะคีย์บอร์ดแบบนี้สะดวกในการเคลื่อนย้าย เช่นการใช้งานในบ้านเป็นต้น แต่อย่างไรก็ตาม ก็มีสายคีย์บอร์ด (เสียเงินเพิ่ม) สำหรับผู้ที่ไม่ชอบการใช้แสงอินฟราเรด หรือการใช้งานในบางสถานที่ เช่นในชั้นเรียนอาจมีการใช้งาน PCjr หลาย ๆ ตัวพร้อม ๆ กัน

ภายในคีย์บอร์ดมีไมโครโพรเซสเซอร์ของ Intel เบอร์ 8048 ซึ่งเป็น CMOS ทำหน้าที่ในการจัดการรับสัญญาณจากคีย์บอร์ดแล้วส่งผ่านไปยังไดโอดอินฟราเรด 2 ตัว ที่จะทำหน้าที่ส่งสัญญาณไปยัง System Unit โดยใช้คลื่นความถี่ขนาด 40 kHz เป็นคลื่นพาหะ ในตัวคีย์บอร์ดจะมีแบตเตอรี่ขนาด AA อยู่ 4 ตัวด้วยกันเป็นตัวจ่ายไฟ ซึ่งจะจ่ายไฟเฉพาะช่วงเวลาสั้น ๆ ที่มีการกดตัวคีย์เท่านั้น จึงประหยัดไฟมาก ทั้งนี้อุปกรณ์ที่กินไฟในคีย์บอร์ดนี้



IBM XT – 370, PC jr, 3270 – PC

มีเพียง 2 ตัว คือ ไมโครโพรเซสเซอร์ 8048 และเครื่องส่งอินฟราเรด

คีย์บอร์ดของ PCjr เป็นแบบปุ่มยาง ซึ่งอาจจะทำให้ผู้ใช้ที่ต้องการพิมพ์สัมผัสรู้สึกรำคาญได้เหมือนกัน เพราะมักมีปัญหา key-rollover กล่าวคือ เมื่อกดคีย์หลายคีย์พร้อม ๆ กัน อักษรบางตัวที่กดอาจจะขาดหายไปได้ แต่สำหรับผู้ใช้ที่เริ่มต้นและเด็กอาจจะชอบก็ได้ เพราะเหมาะและทนทานได้ดีสำหรับการเล่นเกมส์

อุปกรณ์อินฟราเรดสามารถทนทานต่อการรบกวนจากแสงสว่างต่าง ๆ ได้ดี เช่น จากแสงอาทิตย์ แสงไฟ รวมทั้งแหล่งกำเนิดแรงดันขนาดสูง ๆเช่น จาก โทรทัศน์ได้อีกด้วย ในกรณีที่เครื่องได้รับการกวนมาก เครื่องจะส่งเสียงร้อง beep ออกมาเป็นการเตือน

ระบบเสียงของ PCjr

ระบบการสังเคราะห์เสียงของ PCjr ได้รับการปรับปรุงให้ดีขึ้นกว่าของIBM PC มากทั้งนี้เพราะการใช้งานที่สำคัญอย่างหนึ่งของ PCjr ก็คือ การเล่นเกมส์ โปรแกรมที่เกี่ยวกับเสียงที่ออกแบบมาใช้ ใช้กับ PC สามารถใช้กับPCjr ได้แต่โปรแกรมที่เกี่ยวกับเสียงที่ออกแบบมาเฉพาะสำหรับ PCjr จะไม่สามารถใช้กับ PC หรือ PC XT ได้ PCjr สามารถสร้างเสียงที่ซับซ้อนที่มีความสมจริงได้ตั้งแต่เสียงจรวด เสียงสัตว์ประหลาด ไปจนถึงเสียงระเบิดซึ่งบรรดาเสียงต่าง ๆ เหล่านี้เป็นเสียงที่มีมากกว่าหนึ่งโน้ตในเวลาเดียวกัน ซึ่งฮาร์ดแวร์กำเนิดเสียงของ PCjr เท่านั้นที่สามารถทำได้

ระบบเสียงทั้งของ PC และ PCjr จะเริ่มต้นจากที่ความถี่ 1.19 MHz สำหรับใน PC แล้ว ความถี่นี้จะถูกป้อนเข้าไปใน programmable counter ซึ่งเรียกกันในหมู่นักออกแบบอิเลคทรอนิคส์ว่า 8253-5 timer chip ซึ่งจะทำการหารความถี่ดังกล่าวลงไปตามตัวเลขที่โปรแกรมจะส่งมา อาทิเช่น ถ้าโปรแกรม BASIC ส่งตัวเลข 2280 มา ตัว timer ก็จะนำตัวเลขนี้มาหารความถี่หลัก คือ 1.19 MHz และได้ความถี่ 523 Hz ซึ่งเมื่อสัญญาณนี้ได้รับการขยายออกลำโพงก็จะได้ยินเสียงซึ่งตรงกับโน้ตตัว middle C ออกมา

ในกรณีของ PC มีข้อจำกัดอยู่ตรงที่สามารถส่งเสียงมาได้ครั้งละโน้ตเท่านั้น นอกจากนั้นยังไม่สามารถควบคุมความดังของเสียงได้อีกด้วย PCjr สามารถเอาชนะข้อจำกัดเหล่านี้ได้ โดยการใช้ชิพพิเศษคือ Texas Instrument SN 76489A ซึ่งเป็นชิพกำเนิดเสียงที่สามารถสังเคราะห์ได้ถึง 4เสียงพร้อม ๆ กัน โดย 3 เสียงแรกเป็นเสียง tone ส่วนอีกเสียงหนึ่งเป็นเสียง noise โดยแต่ละเสียงนั้นยังสามารถกำหนดความดังได้อย่างอิสระอีกด้วย

ด้านหลังของเครื่อง PCjr มีaudiojack ซึ่งผู้ใช้สามารถนำสัญญาณไปเข้าระบบไฮไฟเพื่อขยายออกลำโพง เพื่อคุณภาพเสียงที่ดีขึ้น นอกจากนั้นยังมีปุ่มที่ใช้ต่อกับ TV ได้โดยตรงเพื่อนำเสียงไปออกยังลำโพงของ TV ได้โดยตรงทั้งนี้เพราะ PCjr มี RF modulator ซึ่งเป็นตัวเปลี่ยนให้เป็นสัญญาณโทรทัศน์อยู่ในตัวอยู่แล้ว

ระบบแสดงภาพของ PCjr

PCjr ไม่จำเป็นต้องมี video-display adapter card เหมือนกับ PC หรือ PC XT ทั้งนี้เพราะวงจรวีดีโอได้ถูกสร้างไว้ในแผงวงจรหลักเรียบร้อยแล้วโดยใช้ชิพ MC6845 สำหรับการควบคุมด้านควบคุมสีและกราฟิค นอกจากนั้นยังได้ใช้ชิพ LSI 5220 ซึ่งเป็น video gate arrayซึ่งสามารถทำให้แสดงสีได้มากกว่า mode



System Unit และคีย์บอร์ดไร้สายของ PCjr

Program Name/Version	Operates on IBM PCjr	Comments	Requires DOS 2.10	Requires BASIC Cartridge
Adventure/1.00	yes	cannot set foreground and background colors	no	no
Adventure in Serenia/1.00	yes	colors may vary from IBM PC	no	no
Adventures in Math/1.00	yes		no	yes
Animation Creation/1.00	yes	will run on an IBM PCjr with 64K memory	no	no
APL (A Programming Language)/1.00	no		n.a.	n.a.
Arithmetic Games (Set 1)/1.00	yes	no color on composite monitor	yes	yes
Arithmetic Games (Set 2)/1.00	yes	no color on composite monitor	yes	yes
Asynchronous Communications Support/2.00	no		n.a.	n.a.
BASIC Compiler/1.00	yes	must have sufficient storage for Compile and Link	yes	no
BASIC Primer/1.00	no		n.a.	n.a.
BASIC Programming Development System/1.05	yes	not recommended for IBM PCjr; use same procedure to load DOS 2.10 as for DOS 2.00	yes	no
Binary Synchronous 3270 Emulation/1.00	no		n.a.	n.a.
BPI Accounting Software (all)	no		n.a.	n.a.
Bumble Games/1.00	yes		no	yes
Bumble Plot/1.00	yes		no	yes
Casino Games/1.05	yes		no	yes
COBOL Compiler/1.00	no	compiler output will run on IBM PCjr if there is sufficient storage	n.a.	n.a.
Decathlon/1.00	no		n.a.	n.a.
Diskette Librarian/1.00	yes		yes	yes
Dow Jones Reporter/1.00	yes		yes	yes
Easywriter/1.15	yes		yes	no
Fact Track/1.00	no		n.a.	n.a.
Filecommand/1.00	yes	use same procedure to load DOS 2.10 as for DOS 2.00	yes	no
Fixed Disk Organizer/1.00	no		n.a.	n.a.
FORTRAN Compiler/1.00	no	compiler output will run on IBM PCjr if there is sufficient storage	n.a.	n.a.
Home Budget, jr/1.00	yes		yes	yes
Homeword/1.00	yes		no	no
Juggles' Butterfly/1.00	yes		no	yes
Learning DOS 2.00/1.00	no		n.a.	n.a.
Learning To Program in BASIC/1.00	no		n.a.	n.a.
Logo/1.00	yes		yes	no
Macro Assembler/1.00	yes		yes	no
Mailing List Manager/1.00	no		n.a.	n.a.
Monster Math/1.00	yes		no	yes
Multiplan/1.10	yes		yes	no
Multiplication Tables/1.00	no		n.a.	n.a.
One-Hundred-and-One Monochrome Mazes/1.00	no		n.a.	n.a.
Pascal Compiler/1.00	no	compiler output will run on IBM PCjr if there is sufficient storage	n.a.	n.a.
Peachtext/1.00	yes	not recommended for IBM PCjr	yes	no
Peachtree Accounting Software (all)	no		n.a.	n.a.
Personal Communications Manager/1.00	yes		no	no
Personal Editor/1.00	yes		yes	no
pfs:FILE/1.05	yes		yes	no
pfs:REPORT/1.05	yes		yes	no
Private Tutor/1.00	no		n.a.	n.a.
Professional Editor/1.00	yes		yes	no
SNA 3270 Emulation/RJE Support/1.00	no		n.a.	n.a.
Strategy Games/1.05	yes		no	yes
3101 Emulation/1.00	no		n.a.	n.a.
Time Manager/1.05	yes		yes	no
Turtle Power/1.00	yes		no	no
Typing Tutor/1.00	no		n.a.	n.a.
UCSD p-System (all products)	no		n.a.	n.a.
Visicalc/1.20	yes		yes	no
Word Proof/1.00	yes		yes	no

Table 2: *IBM PCjr software compatibility chart.*

ตารางที่ 1 แสดง Compatibility ของโปรแกรมของ PC ที่นำมาใช้ใน PCjr

ธรรมดาของ PC

สิ่งที่แตกต่างอย่างสำคัญของโครงสร้างทางด้านวิดีโอของ PC และ PCjr คือการที่ PCjr จำต้องใช้ส่วนหนึ่งใน system memory ในการเก็บ video-display data (PC เก็บไว้ใน display card แทน) ดังนั้นระบบวิดีโอและโพรเซสเซอร์ 8088 จำต้องใช้ Memory ส่วนนี้ร่วมกัน โดยเนื้อที่นี้อาจจำเป็นต้องใช้ตั้งแต่ 16 K สำหรับเป็น display buffer ไปจนถึง 32 K สำหรับใน high resolution mode

การที่ต้องแบ่งการใช้เนื้อที่ใน Memory กันใช้งานนี้เองทำให้ตัวโพรเซสเซอร์ทำงานช้าลงทั้งนี้เพราะ 8088 ต้องรอทุก ๆ 2 ใน 3 ของสัญญาณนาฬิกาเพื่อรอให้วงจรวิดีโอทำการ refresh และออกจอ จากการทดสอบด้วยโปรแกรม benchmark PCjr ทำงานช้ากว่า PC ประมาณ 24-32%

PCjr ไม่มีวงจรที่เลียนแบบการทำงานของ Monochrome Display Adapter ของ PC ดังนั้นจึงไม่อาจจะต่อกับ Monochrome Display ของ IBM ได้ แต่สามารถต่อได้กับจอภาพที่ใช้สัญญาณ RGB (แบบสัญญาณ TV ธรรมดา) แต่จอ RGB ของ IBM ก็มีราคาค่อนข้างแพง (ราคากว่าครึ่งของราคาเครื่อง) ซึ่งผู้ที่คิดจะใช้ระบบนี้ควรจะคิดดูให้ดีเสียก่อน

System Software

PCjr มีภาษา Cassette BASIC มาด้วยใน ROM ของเครื่อง นอกจากนั้นยังมี BASIC รุ่นใหม่คือ Cartridge BASIC ซึ่งเหมือนกับ Advanced BASIC (BASICA) ของ IBM แต่ได้มีการเพิ่มคำสั่งด้านกราฟิคและเสียงตลอดจนได้เพิ่มการใช้งานของ light pen และ joystick เข้าไปอีกด้วย

Software Product	Price
Monster Math	$30
Animation Creation	$40
Mouser	$35
Scubaventure	$35
Homeword*	$75
Crossfire	$35
Mine Shaft	$35
Bumble Games*	$40
Juggles' Butterfly*	$35
Bumble Plot*	$40
Personal Communications Manager	$100
IBM PCjr BASIC (cartridge)	$75
Turtle Power	$50
Fixed Disk Organizer	$50
Adventures in Math	$35
DOS 2.10	$65
Home Budget, jr	$45
Casino Games 1.05	$35
Strategy Games 1.05	$30
Easywriter 1.15*	$175
pfs:FILE 1.05*	$140
pfs:REPORT 1.05*	$125
Time Manager 1.05*	$100
BASIC Program Development System 1.05	$130

*Prices will vary at authorized IBM Personal Computer dealers.

PCjr ใช้ IBM PC-DOS 2.1 ซึ่งเป็น DOS ที่ได้รับการปรับปรุงขึ้นจาก DOS รุ่นก่อน ๆ PC-DOS 2.1 กินเนื้อที่ 24,688 ไบท์ ในขณะที่ DOS 2.0ใช้เพียง 24,576 ไบท์เท่านั้น PC DOS 2.1 สามารถใช้ run ได้ใน PC และPC XT แต่ PCjr ใช้ได้แค่PC DOS 2.1 เท่านั้น ทั้งนี้เพราะใช้ disk-interface routines เฉพาะสำหรับ PCjr เท่านั้น (PCjr ใช้disk ชนิด haft height)

คำสั่ง MODE ใน PC-DOS 2.1 สามารถใช้ได้กว้างกว่าใน DOS 2.0 โดยเฉพาะเมื่อใส่ expansion card ขนาด 64 K เข้าไป (รายละเอียดดูได้จากตาราง) จอภาพของ PCjr จะแสดงเป็น 40คอลัมน์เสมอถ้าไม่สั่งเป็นอย่างอื่นแต่ก็สามารถเปลี่ยนให้เป็น 80 คอลัมน์ได้โดยใช้คำสั่ง MODE 80

แผ่น system disk ของ PC-DOS 2.1 มี file ชื่อ BASICC.COM และ BASICA.COM เหมือนกับใน DOS 2.0 เหมือนกัน แต่มีไว้สำหรับใช้ในเครื่อง PC และ PC XT สำหรับใน PCjr นั้นจำต้องใช้ภาษา BASIC ใน ROM หรือที่เรียกว่า Cartridge BASIC นั่นเอง

Mode	Number of Colors	Compatible with Color/Graphics Adapter	Requires 64K-byte Memory and Display Expansion Board	Buffer Size (Bytes)
40-column alphanumeric	16	yes	no	2K
160 by 200	16	no	no	16K
320 by 200	4	yes	no	16K
640 by 200	2	yes	no	16K
80-column alphanumeric	16	yes	yes	4K
320 by 200	16	no	yes	32K
640 by 200	4	no	yes	32K

Table 3: *PCjr video modes.*

ตารางที่ 2 แสดง Mode ต่าง ๆ ที่เพิ่มขึ้นจากการใช้ PC DOS 2.1 เทียบกับการใช้ Color/Graphic Adapter ใน IBM PC Memory Map ของ PCjr

Memory Area	Address	
	FFFFF	Cartridge Chip Selects
BIOS/Diagnostic/Cassette Basic Program Area	F0000	
Standard Application Cartridge	E8000	
Standard Application Cartridge	E0000	
Reserved For Future Cartridge	D8000	
Reserved For Future Cartridge	D0000	
Reserved for I/O ROM	C0000	
Video RAM	B8000	
Reserved Future Video	A0000	
Reserved Future User RAM	20000	
Expansion RAM	10000	
Base RAM	00000	

การขยายระบบ

IBM ได้แจ้งให้ผู้ใช้ PCjr ทราบแล้วว่า PCjr จะไม่มีข้อจำกัด ในการขยาย Memory หรือ disk storage ดังจะเห็นได้จากรูป memory map ของ PCjr แล้วว่ามี RAM ขนาด 512 K ถูก reserved เอาไว้สำหรับการใช้งานในอนาคตแล้ว

อย่างไรก็ตามการขยายระบบไม่ว่าจะเป็น RAM หรือ disk storage ต้องการไฟฟ้ามากขึ้นและในตัว PCjr เองก็ไม่มีที่พอ ดังนั้นจำเป็นต้องมี expansion box ที่มี power supply ต่างหาก โดยต่อเชื่อมโยงกันทาง I/O expansion bus ขนาด 60 pin ปัจจุบันมีผู้ทำ expansion box ที่บรรจุ floppy disk drive หรือ harddisk พร้อมด้วย expansion slot อีก 4 slot ออกมาจำหน่ายแล้ว แต่การขยายระบบด้วย expansion unitก็นับว่าเป็นการลงทุนด้วยราคาที่ค่อนข้างแพงเมื่อเทียบกับการขยายระบบจาก PC หรือ PC XT

PC VS PCjr

IBM PCjr ได้รับการออกแบบให้สามารถใช้โปรแกรมของ IBM PC และ PC XT ได้มากที่สุด แต่อย่างไรก็ตามจากขนาดความจำที่จำกัด ระบบ video display และ sound ที่ได้รับการดัดแปลง ตลอดจนข้อแตกต่าง ต่าง ๆ ทางฮาร์ดแวร์เล็ก ๆ น้อย ๆ อื่น ๆ ทำให้ PCjr ไม่สามารถที่จะ compatible กับ IBM PC ได้ 100% ตารางที่ 1 จะเป็น Compatibility chart แสดงถึงการใช้งานของโปรแกรมสำคัญต่าง ๆ ของ PC และข้อแตกต่างที่สำคัญ ๆ ที่เกิดขึ้นเมื่อนำมาใช้กับ PC jr.

สรุป

IBM PCjr เป็นผลิตภัณฑ์หนึ่งของ-IBM ที่ได้ออกมาเสริม Production line ของตนเพื่อเสริมสร้างอาณาจักร PC ของตนให้เข้มแข็งขึ้น และเป็นตัวแรกที่ลงมาในตลาด home computer PCjr อาจจะเป็น home computer ที่มีราคาแพงเครื่องหนึ่ง (ในอเมริกา $ 1250 สำหรับรุ่นที่มี diskdrive) แต่ถ้านับว่าเป็น Personal computer ราคาก็อยู่ในระดับพอสมควร

สำหรับข้อจำกัดของ pcjr เช่น Memory ที่จำกัด คีย์บอร์ดที่เป็นปุ่มยาง ตลอดจนการมี drive เดียวปัจจุบันข้อจำกัดเหล่านี้ก็กำลังจะถูกจำกัดไปทีละข้อแล้วโดยบริษัทฮาร์ดแวร์ต่าง ๆ ที่สนับสนุน IBM อยู่

ถึงแม้ว่าซอฟท์แวร์ของ PCjr จะไม่ Compatible 100% กับของ PC แต่จำนวนซอฟท์แวร์ที่ใช้ใน PCjr ได้นั้นก็ยังมีจำนวนมากมายมหาศาลและยังเป็นการเปิดตลาดทางด้านเกมส์และโปรแกรมการศึกษาสำหรับเครื่อง IBM ขึ้นมาอีกทางหนึ่งด้วย

IBM PCjr จำต้องใช้เวลาสักระยะหนึ่งในการพิสูจน์ตัวเองในตลาดว่าจะประสบความสำเร็จได้อย่าง IBM PC เครื่องรุ่นพี่ได้หรือไม่และเส้นทางของ PCjr ก็ยังมีอีกไกลนักในตลาด home computer ที่เต็มไปด้วยเสือสิงห์กระทิงแรด เราน่าจะลองคอยดูว่า PCjr จะเป็นดาวรุ่งหรือลูกเป็ดขี้เหร่กันแน่ และจะไปได้ไกลกันได้สักแค่ไหนในยุทธจักรไมโครคอมพิวเตอร์ ■

เสรีวัฒน์

และแล้วก็ถึงยุคของ.... Integrated Software

เดี๋ยวนี้ใคร ๆ ในวงการคอมพิวเตอร์ต่างก็พูดกันถึง Integrated Software วิวัฒนาการใหม่ของซอฟท์แวร์ที่ทำให้โปรแกรมหลายโปรแกรมทำงานด้วยกันได้ทั้งนี้เป็นผลมาจากการพัฒนาด้านฮาร์ดแวร์ที่ทำให้หน่วยความจำมีขนาดใหญ่และโปรเซสเซอร์มีความเร็วสูงขึ้นจนสิ่งต่าง ๆ เหล่านี้เป็นจริงขึ้นได้

ซอฟท์แวร์เหล่านี้มีหลายประเภทด้วยกันนับตั้งแต่โปรแกรมโปรแกรมเดียวที่ทำงานได้หลาย ๆ งานไปจนกระทั่ง Operating System ที่ทำให้โปรแกรมหลาย ๆ โปรแกรมทำงานพร้อม ๆ กันได้ ตลอดจนถึงโปรแกรมหลายโปรแกรมที่ใช้ระบบฐานข้อมูล (database) เดียวกัน

คำว่า Integrated Software เป็นคำใหม่ที่เริ่มจะได้ยินกันในปีนี้คล้าย ๆ กับคำว่า "User-frieldly" หรือโปรแกรมที่เป็นมิตรกับผู้ใช้ซึ่งเริ่มฮิตเมื่อปีที่แล้วในทั้ง 2 กรณีนี้ ก็เป็นสิ่งบอกถึงการตอบสนองของอุตสาหกรรมซอฟท์แวร์ที่มีต่อผู้ใช้คอมพิวเตอร์นั่นเอง

ผู้ใช้คอมพิวเตอร์ได้บ่นและเรียกร้องกันมานานแล้วถึงโปรแกรมคอมพิวเตอร์ที่ง่ายต่อการใช้และการเรียนรู้และในปัจจุบันบริษัทซอฟท์แวร์ต่าง ๆ ก็ได้เริ่มตอบสนองข้อเรียกร้องอันนี้แล้ว

แต่ว่าความฝันของผู้ใช้คอมพิวเตอร์จะเป็นจริงได้นั้นก็ต้องรอกันจนกระทั่งถึงยุคของไมโครคอมพิวเตอร์ 16 บิท แต่เดิมนั้นเป็นยุคของคอมพิวเตอร์ 8 บิท ซึ่งมี Memory ได้ไม่เกิน 64K ปัจจุบันนี้ไมโครคอมพิวเตอร์รุ่นใหม่สามารถมี RAM มากกว่าเดิมได้ถึง 10 เท่าและนอกจากนั้น อุปกรณ์ชี้ตำแหน่ง (pointing device) จอภาพที่มีกราฟิคและ Winchester hard-disk drive ที่มีขนาดตั้งแต่ 5,10 หรือ 15 เมกกะไบท์ได้กลายเป็นของธรรมดาไปเสียแล้ว สิ่งเหล่านี้เองได้เป็นสิ่งที่แผ้วถางทางให้ซอฟท์แวร์รุ่นใหม่ที่ทำงานได้หลาย ๆ ฟังชั่นอย่างที่ทุก ๆ คนต้องการ Integrated Software นั้นมีอยู่หลายรูปแบบด้วยกัน สำหรับผู้ผลิตบางรายแล้ว Integrated Software หมายถึงชุดของคอมพิวเตอร์โปรแกรมหลาย ๆ โปรแกรมที่ทำงานด้วยกันได้เท่านั้น บางรายก็นิยามไว้ว่าคือโปรแกรมโปรแกรมเดียวที่ทำงานได้หลาย ๆ อย่าง บางรายก็หมายถึง Operating environment แบบใหม่ซึ่งโปรแกรมหลาย ๆ โปรแกรมจะสามารถทำงานด้วยกันได้ หรือบางรายก็นิยามว่าเป็นชุดของโปรแกรมที่ใช้ระบบฐานข้อมูลเดียวกัน จะเห็นได้ว่าผู้ตีความ Integrated Software หลายรูปแบบด้วยกันแต่ทั้งนี้ต่างมีจุดประสงค์อย่างเดียวร่วมกันคือต่างก็ต้องการให้คนใช้คอมพิวเตอร์ได้ง่ายขึ้น

โปรแกรมที่ทำงานประสานกัน

รูปแบบของ Integrated Software ที่ง่ายที่สุดนั้นก็คือชุดของโปรแกรมที่ออกแบบมาให้ทำงานคล้าย ๆ กันและใช้ไฟล์ข้อมูลร่วมกัน ชุดของโปรแกรมดังกล่าวมักจะมาจากผู้ผลิตซอฟท์แวร์เดียวกัน จึงจะสามารถใช้ฟังก์ชั่นหรือคำสั่งที่ เหมือนกันหรือคล้ายกันได้ในทุกโปรแกรม การใช้งานทำได้ง่ายเพราะไม่ต้องจดจำคำสั่งมาก

เนื่องจากโปรแกรมทุกโปรแกรมของผู้ผลิตเดียวกันมักจะมี format ของ file ที่เหมือนกันดังนั้นการส่งผ่านข้อมูลระหว่างโปรแกรมเหล่านั้นจึงทำได้ง่าย แต่อย่างไรก็ตามการที่จะได้ระบบโปรแกรมดังว่านี้มาคุณจำเป็นต้องซื้อโปรแกรมทั้งชุดจากผู้ผลิตเดียวเท่านั้น เพื่อเป็นหลักประกันว่าโปรแกรม

เหล่านั้นจะทำงานได้จริง ๆ และแต่อย่างไรก็ตามโปรแกรมเหล่านั้นยังทำงานพร้อม ๆ กันจริงไม่ได้อยู่นั่นเอง

ชุดของโปรแกรมที่รู้จักกันดีที่ Integrate แบบนี้ได้แก่ชุดของ Perfect Software อันได้แก่ Perfect Writer Perfect Calc, Perfect Speller และ Perfect Filer

โปรแกรมแบบทำงานหลายฟังชั่น

โปรแกรมชนิดนี้กำลังเป็นที่นิยมและเป็นที่กล่าวขวัญถึงเป็นอย่างมากในขณะนี้โดยเฉพาะอย่างยิ่งโปรแกรม 2 ตัว คือ Lotus 1-2-3 ของ Lotus Corporation และ MBA จาก Context Management System ทั้งสองโปรแกรมทำหน้าที่ได้หลายอย่างในโปรแกรมเดียว อาทิเช่น Lotus 1-2-3 นั้นได้รวมเอาโปรแกรม spread sheet, graphic และ database ไว้ด้วยกัน ในขณะที่ MBA ยังมีโปรแกรม-communication และ Wordprocessing เพิ่มขึ้นมานอกจากของ Lotus นอกจากนั้นทั้งสองโปรแกรมยังให้คุณสามารถแยกหน้าจอออกเป็นหลาย Window โดยแต่ละ-Window คือโปรแกรมโปรแกรมหนึ่ง

Integrated Operating Environment

แนวความคิดอย่างใหม่ของ Integrated Software คือการสร้าง Operating Environment ขึ้นจาก Operating System แล้วทำให้โปรแกรมหลาย ๆ โปรแกรมทำงานพร้อม ๆ กันได้โดยแต่ละโปรแกรมจะมี Window ของตัวเอง

Apple Lisa เป็นตัวอย่างที่ดีของโปรแกรมระบบนี้ แต่ปัจจุบันได้มีบริษัทซอฟท์แวร์หลายแห่งที่ได้พัฒนาโปรแกรมประเภทเดียวกันนี้ในราคาที่ถูกกว่าสำหรับเครื่อง 16 บิท อย่างเช่นบริษัท Visicorp, Digital Research, Micropro และ Microsoft.

Operating environmment นั้นแตกต่างจาก Multiple-function program ตรงที่คุณสามารถเลือก Application Software ตัวที่คุณชอบมาใช้ได้โดยไม่ถูกจำกัดให้ใช้ฟังชั่นที่มากับโปรแกรม Multi function เท่านั้น

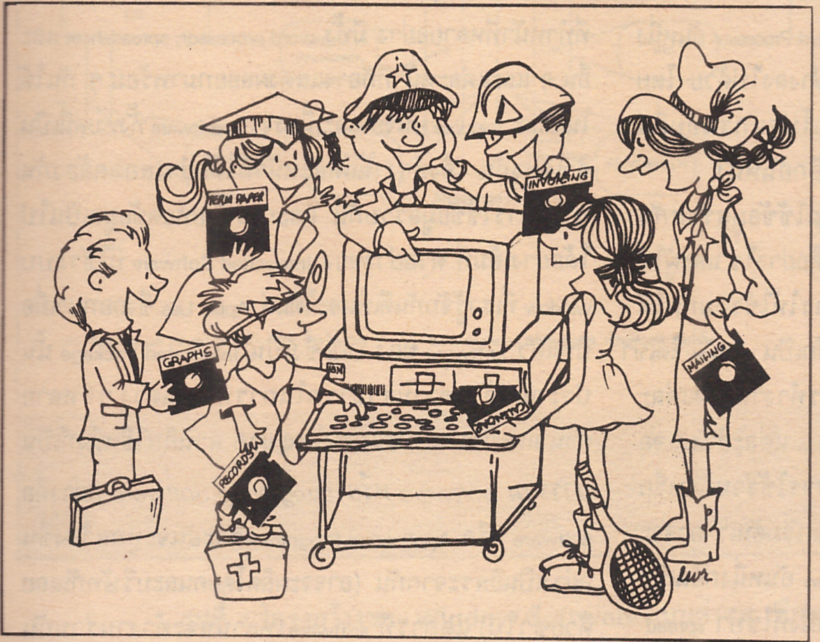
Operating environment ส่วนใหญ่มักจะต้องการระบบคอมพิวเตอร์ที่มีราคาค่อนข้างแพงและมักจะต้องการซอฟท์แวร์ที่ออกแบบมาเฉพาะเพื่อจะใช้จุดเด่นของฮาร์ดแวร์ให้มากที่สุดแต่อย่างไรก็ตามก็ยังเปิดโอกาสให้ผู้ใช้เลือกออกแบบระบบของตนเองได้โดยใช้ Software ตัวที่ตนชอบ

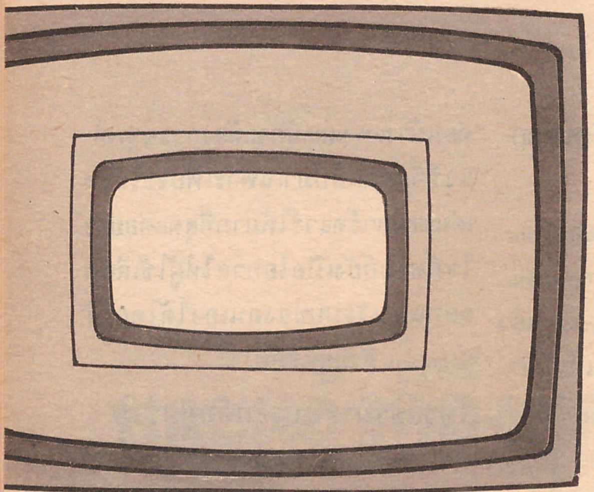
โปรแกรมระบบฐานข้อมูลร่วม

Integrated Software ประเภทนี้เราอาจจะไม่ค่อยคุ้นเคยกับมันนัก เมื่อเร็ว ๆ มานี้มีผู้ผลิตซอฟท์แวร์หลายรายได้แนะนำระบบบริหารงานฐานข้อมูล (database management System) ที่ออกแบบมาให้เป็นแกนให้โปรแกรมใช้งานต่าง ๆ สามารถใช้ข้อมูลได้ โปรแกรมดังกล่าวมักจะมีตัว interface กับโปรแกรมที่เป็นที่นิยมอยู่ด้วยเสมอ บริษัทผู้ผลิตดังกล่าวได้แก่ Condor Computer Corporation, Microrim Inc, Software Products International และ Alpha Software Corporation

แนวความคิดในการที่ให้โปรแกรมทั้งหมดใช้ฐานข้อมูลเดียวร่วมกันนั้นแตกต่างจาก Integrated Operating enviromment ตรงที่มันไม่ได้ทำงานพร้อม ๆ กัน ในขณะเดียวกันมันก็ทำงานคล้ายกับเป็นโปรแกรมหลายฟังชั่นเหมือนกัน แต่ว่าแต่ละโปรแกรมนั้นผู้ใช้สามารถเลือกมาใช้เองได้

ถ้าคุณอ่านบทความนี้แล้วเกิดความรู้สึกว่าตลาดซอฟท์แวร์กำลังเกิดการเปลี่ยนแปลงขนานใหญ่แล้วละก็คุณคิดได้ถูกต้องแล้ว ความเปลี่ยนแปลงนี้ได้เริ่มเกิดขึ้นเมื่อไมโครโพรเซสเซอร์ขนาด 16 บิท Memory ขนาดใหญ่และ harddisk drive ได้เริ่มเข้ามาสู่ตลาด ทำให้ผู้ผลิต Software สามารถสร้างยุคใหม่ของโปรแกรมที่ใช้ได้ง่ายและทรงประสิทธิภาพให้เกิดขึ้นได้ ■





โดย สุวัฒน์

Window Software

ในปี ค.ศ.1984 นี้เราอาจจะกล่าวได้ว่าเป็นปีของการเริ่มต้นยุคใหม่ของระบบ Software โดย**ตัวคอมพิวเตอร์จะมีสภาพเป็นหน้าต่าง (window) ที่เปิดออกไปสู่โลกของข้อมูลในระบบต่าง ๆ** แนวคิดพื้นฐานของมันก็คือทำอย่างไรจึงจะสามารถเชื่อมโยงข้อมูลที่อยู่ในระบบหนึ่ง (โปรแกรมหนึ่ง) เข้ากับระบบข้อมูลอื่น ๆ (โปรแกรมอื่น ๆ ได้) ความสามารถในด้านนี้ของ Software จะทำให้เราจัดการกับข้อมูลที่มีอยู่ที่ไหนก็ตามในระบบได้โดยใช้โปรแกรมต่าง ๆ มาทำงานร่วมกัน โดยที่การทำงานของโปรแกรมต่าง ๆ นั้นต่างก็เป็นอิสระไม่เกี่ยวข้องกัน คือ เราอาจทำการวิเคราะห์ข้อมูลด้วยโปรแกรมอันหนึ่งแล้วนำข้อมูลที่ได้ไปใช้กับอีกโปรแกรมหนึ่งได้ ยกตัวอย่างเช่นเราอาจจะเขียนจดหมายหรือรายงานสักฉบับหนึ่งโดยใช้โปรแกรม Word Processor อันหนึ่ง และเราต้องการที่จะใส่รูปกราฟบางอย่างลงไปด้วย โดยรูปกราฟอันนี้อาจจะถูกสร้างมาจากโปรแกรมอันอื่น เช่น สร้างจากโปรแกรม spreadsheet อีกอันหนึ่ง

เมื่อโปรแกรมต่าง ๆ สามารถใช้ข้อมูลร่วมกันแล้ว ปัญหาที่ตามมาก็คือ จะแสดงผลได้อย่างไร และผู้ใช้จะกำหนดการถ่ายทอดข้อมูลที่ต้องการให้ใช้ร่วมกันได้อย่างไร นั่นเองที่ลักษณะการแบ่งจอออกเป็น window ได้เข้ามามีบทบาทในส่วนนี้ด้วยการแสดงการทำงานของแต่ละโปรแกรม ที่ใช้ข้อมูลร่วมกันบน window แต่ละอันบนจอ ผู้ใช้จะสามาถเห็นข้อมูลทั้งหมดที่ต้องการใช้ร่วมกันหรือถ่ายทอดข้อมูลได้สะดวกขึ้น ตัวอย่างเช่น เริ่มต้นเราอาจจะสร้างจดหมายหรือรายงานนั้นใน window อันหนึ่งก่อนเมื่อเสร็จแล้วเราก็จะไปยัง window อีกอันหนึ่งที่ใช้ทำ spreadsheet แล้วใช้ข้อมูลจาก window อันก่อนมา plot กราฟต่อมาเราจะใช้อุปกรณ์อื่นเช่น mouse ชี้บอกโปรแกรมที่ทำ window ให้แยกกราฟที่ plot ออกมาแล้วส่งผ่านจาก window ที่เป็น spreadsheet ไปยัง window ที่เป็น Word Processor ดังนั้นจดหมายหรือรายงานของเราก็จะมีรูปกราฟจากข้อมูลนั้นด้วยตามต้องการ

Integrated Software VS Operating Environment

การที่ Software ใด ๆ จะมีขีดความสามารถในการแสดงผลในลักษณะของ window หลาย ๆ อันบนจอนั้นอาจทำได้ใน 2 ลักษณะ คือ *ลักษณะแรกเป็น software รวม หรือ Integrated Software* ไปเลย คือ software ตัวเดียวที่ทำหน้าที่หลายอย่าง มีทั้ง word processor, spreadsheet และอื่น ๆ และแต่ละหน้าที่ก็อาจแสดงผลออกมาพร้อม ๆ กันได้ในแต่ละ window บนจอ และเนื่องจาก software ทั้งหมดนี้เป็นตัวเดียวกัน จึงทำงานแต่ละหน้าที่ได้อย่างสอดคล้องกัน ทำให้การใช้ข้อมูลร่วมกัน และการถ่ายทอดข้อมูลเป็นไปได้อย่างเต็มที่ ตัวอย่างของ Integrated Software ที่ใช้ระบบ window ที่เรารู้จักกันดีเห็นจะได้แก่ Apple Lisa ซึ่งออกมาเมื่อปีที่แล้ว software ของ Lisa ซึ่งเป็น Integrated Package นั้นประกอบด้วยความสามารถในการทำงานต่าง ๆ หลายด้าน และมีขนาดโปรแกรมใหญ่มาก ส่วนอีกวิธีหนึ่งก็เป็น *การรวม software หรือ Integrating Software* กล่าวคือ software หรือ Application Program แต่ละอันจะถูกสร้างขึ้นอย่างเป็นอิสระจากกัน (อาจจะผลิตโดยคนละบริษัทกันเลยด้วยซ้ำไป) แต่การที่ software เหล่านั้นจะทำงานร่วมกัน

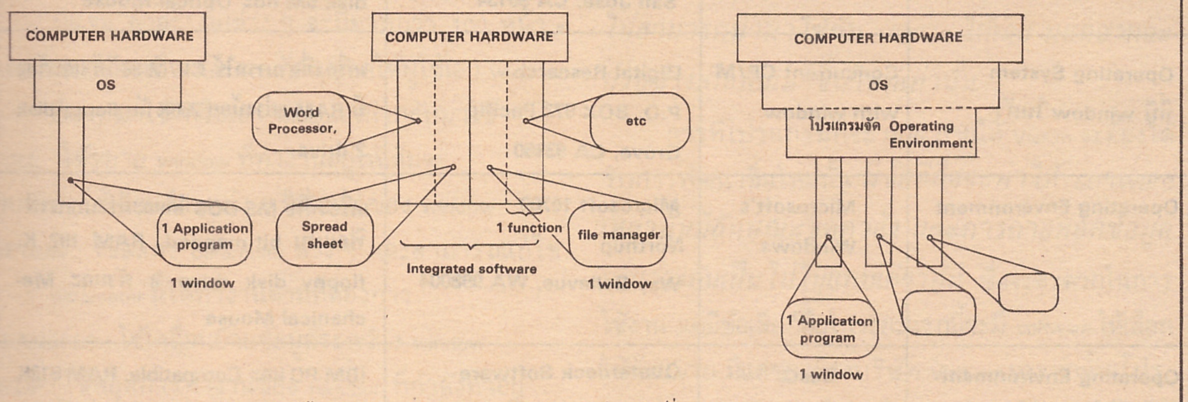
และแสดงผลออกทาง window ของตนเองแต่ละอันบนจอได้นั้น จะต้องมีโปรแกรมอีกอันหนึ่งมาทำหน้าที่จัดการโดยโปรแกรมดังกล่าวจะทำหน้าที่คล้าย ๆ กับโปรแกรมควบคุมระบบ (Operating System-OS) ในการจัดการรับ-ส่งข้อมูลให้กับโปรแกรมแต่ละอัน จัดการถ่ายทอดข้อมูลจัดการแสดงผลทาง window พูดรวม ๆ ก็คือจัดสภาพแวดล้อมในการทำงาน (Operating Environment) ให้สามารถใช้ window ได้นั่นเอง

การที่จะจัด Operating Environment ให้ใช้ window ได้นั้นก็อาจทำได้ใน 2 ลักษณะอีกเช่นกัน คือรวมลักษณะการทำงานแบบ window เข้าใน OS เลย (เช่นใน Concurrent CP/M version ใหม่ที่ชื่อ Concurrent DOS) หรือใช้โปรแกรมอีกอันหนึ่งทำหน้าที่จัด Operating Environment สำหรับระบบ window โดย RUN ภายใต้ OS อีกทีหนึ่ง (ในทางปฏิบัติแล้ว โปรแกรมดังกล่าวมักจะ Overlay หรือแทนที่บางส่วนของ OS ไปเลย)

ในการใช้งานคอมพิวเตอร์ที่แล้ว ๆ มานั้น การเขียน Application Program ต่าง ๆ อาจจะไม่ต้องอาศัยความสามารถของ window เลยก็ได้ ถ้าเราอยากจะทำจดหมาย เราก็เลือก

ตัวคอมพิวเตอร์จะมีสภาพเป็นหน้าต่าง (window) ที่เปิดออกไปสู่โลกของข้อมูลในระบบต่าง ๆ

หา Software ดี ๆ ที่ทำ Word Processor ก็ได้และคนที่ใช้คอมพิวเตอร์ที่ใช้มันทำงานต่าง ๆ โดยเฉพาะอย่างไม่ปนกันในเวลาเดียวกัน แต่ในอนาคตอันใกล้นี้ ความสามารถที่จะใช้ข้อมูลจากโปรแกรมต่าง ๆ ได้หลายโปรแกรมในขณะเดียวกัน จะได้เปรียบและมีประสิทธิภาพมากกว่า และในปัจจุบัน software ที่ใช้ระบบ window ก็กำลังได้รับความสนใจอย่างมาก ที่เป็นเช่นนี้ก็เพราะ *ไมโครคอมพิวเตอร์ยุคปัจจุบันมีประสิทธิภาพสูงมากพอเพียงที่จะรับภาระการทำงานอันยุ่งยากซับซ้อนนี้* และมี Memory



ก.) Application Program 1 ตัว RUN ภายใต้ OS

ข.) Integrated Software 1 ตัวแต่หลายฟังก์ชั่น

ค.) Application Program หลายตัว RUN ภายใต้โปรแกรมจัด Operating Environment ซึ่ง RUN ใต้ OS หรือเป็นส่วนหนึ่งของ OS แล้วแต่กรณี

รูปที่ 1 โครงสร้างของ Window Software แบบต่าง ๆ เทียบกับ Software ธรรมดา

มากพอ ราคาก็ไม่สูงเกินไป ทำให้ไมโครคอมพิวเตอร์ที่ใช้ระบบ window นี้กลายเป็นสิ่งปรกติธรรมดาไปได้

ในระบบ window นี้เครื่องที่ใช้ต้องมีความสามารถสูงพอควร การเข้าสู่ยุคของเครื่องระบบ 16 บิท และ 32 บิท โดยมีไมโครโปรเซสเซอร์ที่มีความสามารถและประสิทธิภาพสูงกว่าแบบ 8 บิท มาก ทั้งในด้านโครงสร้างภายในและความสามารถในการอ้าง address ที่เพิ่มขึ้น ทำให้สามารถทำงานที่ซับซ้อนซึ่งในระบบ 8 บิท ทำไม่ได้

ราคาของหน่วยความจำที่ลดลงอย่างมาก ก็เป็นเหตุผลที่สำคัญอีกอย่างหนึ่งที่ทำให้ระบบ window กลายเป็นความจริงได้ เพราะข้อมูลและตัวโปรแกรมที่ซับซ้อนและผลที่ได้จำเป็นต้องใช้หน่วยความจำมากพอสมควรในการเก็บในทุกวันนี้เครื่องคอมพิวเตอร์ของเราสามารถที่จะมีหน่วยความจำได้เป็นร้อย ๆ K byte ทีเดียวและเมื่อมีหน่วยความจำมากพอในราคาไม่แพง การแสดงผลบนจอภาพแบบ High-resolution หรือความละเอียดสูง ๆ ก็จะทำได้โดยการใช้เทคนิค bit-mapped display (bitmapped display จะใช้หน่วยความจำมากเป็นพิเศษ)

คุณเคยรู้สึกไหมว่า จอคอมพิวเตอร์เพียงจอเดียวนั้นช่างไม่พอกับการทำงานของคุณเสียเลย

ในปัจจุบันนี้ มีผู้ผลิตซอฟท์แวร์หลายรายกำลังเร่งนำผลิตภัณฑ์ window software ออกสู่ตลาด ซึ่งที่เราจะเสนอในบทความนี้ก็เป็นเรื่องราวโดยสังเขปของ software ของทางบริษัทที่น่าสนใจดังนี้

window software อันแรกที่เราจะกล่าวถึงก็คือ VisiOn ผลิตโดย บริษัท VisiCorp ผู้ผลิต VisiCalc ที่ทำให้เครื่องคอมพิวเตอร์ ดังเป็นพลุและขายดีมาจนทุกวันนี้

VisiOn

ที่เราจะพูดถึง VisiOn ก่อนนี้ก็เพราะว่าเราอาจจะถือ

ประเภท	ชื่อโปรแกรม	ผู้ผลิต	ใช้กับเครื่อง
Operating Environment	Vision	VisiCorp 2895 Zanker Rd. San Jose, CA 95134	IBM PC และ Compatible พร้อม graphic adapter board, RAM 512 K, Hard disk 5M และ Optical Mouse
Operating System ที่มี window ในตัว	Concurrent CP/M with window	Digital Research P.O. BOX 573 Pacific Grove, CA 93950	เครื่องที่สามารถใช้ CP/M-86 ได้ และต้องมี RAM อย่างน้อย 256K กับ floppy disk 2 drive
Operating Envernment	Microsoft's Windows	Microsoft 10700 Northup Way Bellevue, WA 998004	เครื่องที่ใช้ MS-DOS ได้และมีระบบกราฟฟิคแบบ bit-mapped, RAM 192 K, floppy disk drive 2 ตัวและ Mechanical Mouse
Operating Environment	DesQ	Quaterdeck Software 1918 Main St., Suite 240 Santa Monica, CA 90405	IBM PC และ Compatible, RAM 512K และ Hard disk 5 M
Integrated Software	Jane	Arktronics Corp. 113 S. Fourth Ave. Ann Arbon, MI 48104	Apple II, II+, IIe กับ RAM 64K Commodore 64

ตารางที่ 1.

ได้ว่า ระบบ window นี้ VisiOn เป็น Package ตัวแรกที่เกิดขึ้นมาก่อนใคร สำหรับงานบริหารและงานธุรกิจ VisiOn เป็นระบบรวมที่สมบูรณ์มากทำให้คอมพิวเตอร์ส่วนตัวมีจอเป็น window ที่จะให้ผู้ใช้ทำงานหลาย ๆ อย่างขณะเดียวกันบนจอเดียวคล้ายกับกระดาษทดบนโต๊ะหรือสมุดบนโต๊ะทำงาน คือ window ต่าง ๆ จะนำมาเรียงซ้อนกัน (ทับกันแบบ Stacked คือ window อันแรกสุดจะอยู่ในสุด) window ที่นำเสนองานที่กำลังทำจะอยู่บนสุดและมีเนื้อที่บนจอมากกว่า window ของงานอื่น ๆ ระบบ mouse จะช่วยให้ผู้ใช้จัดการเกี่ยวกับข่าวสารข้อมูลภายในแต่ละ window และส่งผ่านข่าวสารข้อมูลระหว่าง window ด้วย สำหรับ mouse ของ VisiOn แตกต่างจาก mouse ทั่ว ๆ ไปบ้าง โดย VisiOn mouse ใช้ระบบ sensor แบบ infrared ในการเลือกตำแหน่งในขณะที่ mouse อื่น ๆ ใช้ระบบ mechanical ในการ sense การเคลื่อนไหวของ mouse การทำงานของ VisiOn mouse จะให้ความแม่นยำและเชื่อถือได้มากกว่า ในระบบ window บางระบบจะใช้รูปภาพช่วย รูปภาพเหล่านี้จะอยู่บนจอเพื่อบอกถึง function ต่าง ๆ (เหมือน Apple Lisa และ Macintosh) ถ้าใช้ mouse ชี้ไปยังรูปภาพใด ๆ เครื่องก็จะทำงานตาม function นั้น ๆ เช่นชี้ไปที่รูป floppy disk ก็อาจจะทำให้มีการแสดง menu ของการใช้งานเกี่ยวกับ disk ก็ได้ (เช่น copy delete) ใน VisiOn ไม่ได้ใช้รูปอย่างที่ว่า แต่จะใช้คำแทนตรงบริเวณล่างสุดของจอจะมีคำสั่งอยู่ 8 คำสั่ง คือ

HELP : เป็น function ที่จะช่วยอธิบายสิ่งต่าง ๆ ที่ปรากฏบนจอ คำอธิบายต่าง ๆ จะมีมากกว่า 100 หน้า

OPEN, CLOSE : เป็น function ที่อนุญาตให้ผู้ใช้เอา window อื่น ๆ มาใช้ หรือเอา window เก่ากลับมาใช้

FULL : ใช้ขยาย window ที่ทำงานอยู่ให้เต็มจอ

FRAME : ใช้เลื่อนตำแหน่ง หรือขนาดของ window

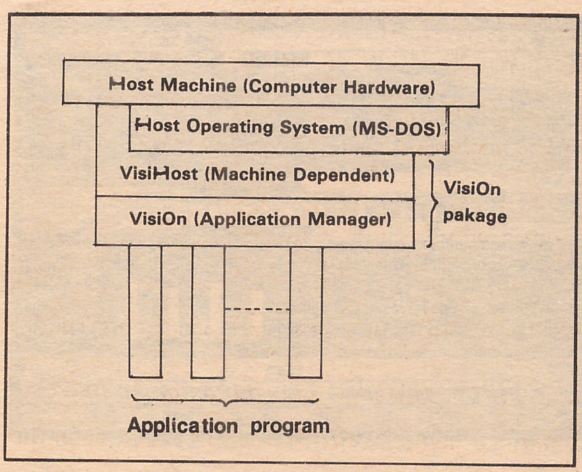
OPTION : ใช้แสดง parameter ต่าง ๆ และสามารถแก้ไข parameter นี้ให้ทำงานตามที่ต้องการ

TRANSFER : ใช้เคลื่อนย้ายข้อมูลระหว่าง window

STOP : ใช้หยุดการทำงานของ function ที่กำลังทำอยู่

VisiOn นี้จะทำตัวเป็นชั้น หรือเปลือก (Shell) คือเป็นโปรแกรมอิสระที่ทำงานร่วมกับ MS-DOS ซึ่งเป็น OS ได้และ RUN โปรแกรมต่าง ๆ โดย VisiOn จะเป็น "ผู้จัดการ" (Application Manager) คอยหุ้มหรือควบคุมโปรแกรมต่าง ๆ อีกทีหนึ่ง

โครงสร้างของ VisiOn อาจอธิบายได้ดังในรูปต่อไปนี้

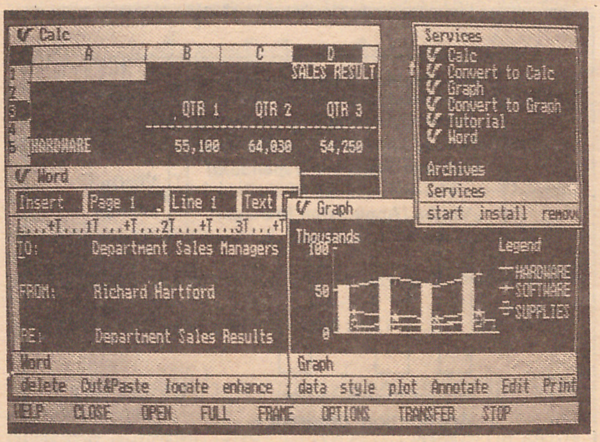


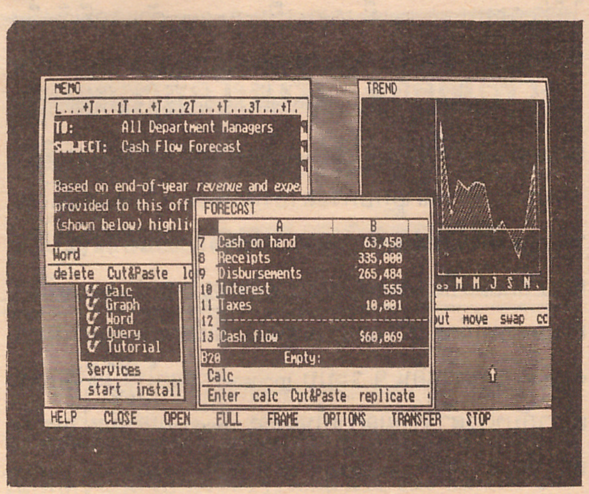
รูปที่ 2 โครงสร้างของ VisiOn มีทั้งส่วนที่ Dependent และ Independent กับ Hardware

เครื่องคอมพิวเตอร์ที่ใช้ RUN VisiOn เปรียบเสมือนเจ้าภาพหรือ Host โดย VisiOn จะ RUN ภายใต้ Host Operating System คือ HS-DOS แต่ตัว VisiOn นี้ที่จริงแล้วยังแบ่งเป็น 2 ส่วน คือส่วนที่เป็น VisiHost ซึ่งจะทำงานร่วมกับ HS-DOS และบางส่วนก็จะควบคุม Computer Hardware โดยตรง VisiHost นี้จะขึ้นกับลักษณะของฮาร์ดแวร์ของคอมพิวเตอร์แต่ละเครื่อง (Machine dependent) อีกส่วนหนึ่งคือ VisiOn จริง ๆ จะทำงานร่วมกับโปรแกรมใช้งานต่าง ๆ ที่นำมา RUN ซึ่งโปรแกรมเหล่านี้จะต้องเข้ากันกับ VisiOn ได้ ดังนั้น *โปรแกรมใช้งานทั่วไปที่ RUN ภายใต้ MS-DOS ได้ จะไม่สามารถนำมาใช้กับ VisiOn ได้เลย นอกจากจะปรับปรุงหรือพัฒนาขึ้นใหม่เท่านั้น*

การที่โปรแกรมที่จะ RUN ภายใต้ VisiOn จะต้องได้รับการพัฒนาขึ้นใหม่นั้น *มีข้อดีคือจะทำให้ software เหล่านั้นมีลักษณะชุดคำสั่ง และการถ่ายทอดข้อมูลระหว่างกันเป็นไปในแบบเดียวกัน* ซึ่งจะสะดวกในการใช้งาน แต่มีข้อเสียก็คือในระยะแรกนี้จะมี software ให้เลือกใช้น้อยมาก มีแต่ของของบริษัท VisiCorp เองเท่านั้น โปรแกรมความสำเร็จรูปหรือ package ที่คุณเคยใช้อยู่จะใช้ไม่ได้เลย แต่บริษัท VisiCorp ก็ได้พยายามชักจูงและเสนอจะให้ความสนับสนุนบริษัทอื่น ๆ ที่จะพัฒนา software ขึ้นใช้ร่วมกับ VisiOn นี้อย่างมาก

สำหรับในตอนนี้ระบบที่ใช้ VisiOn ก็มี IBM PC และ XT โดยบริษัท IBM และบริษัท VisiCorp ก็ได้ลงนามตกลงกันให้ IBM ขาย VisiOn และ software อื่น ๆ ของ VisiCorp

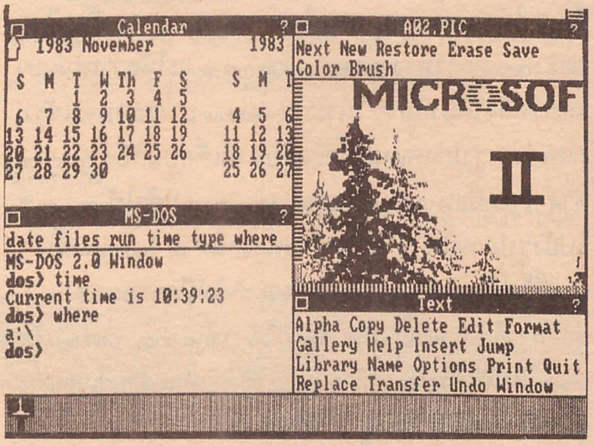


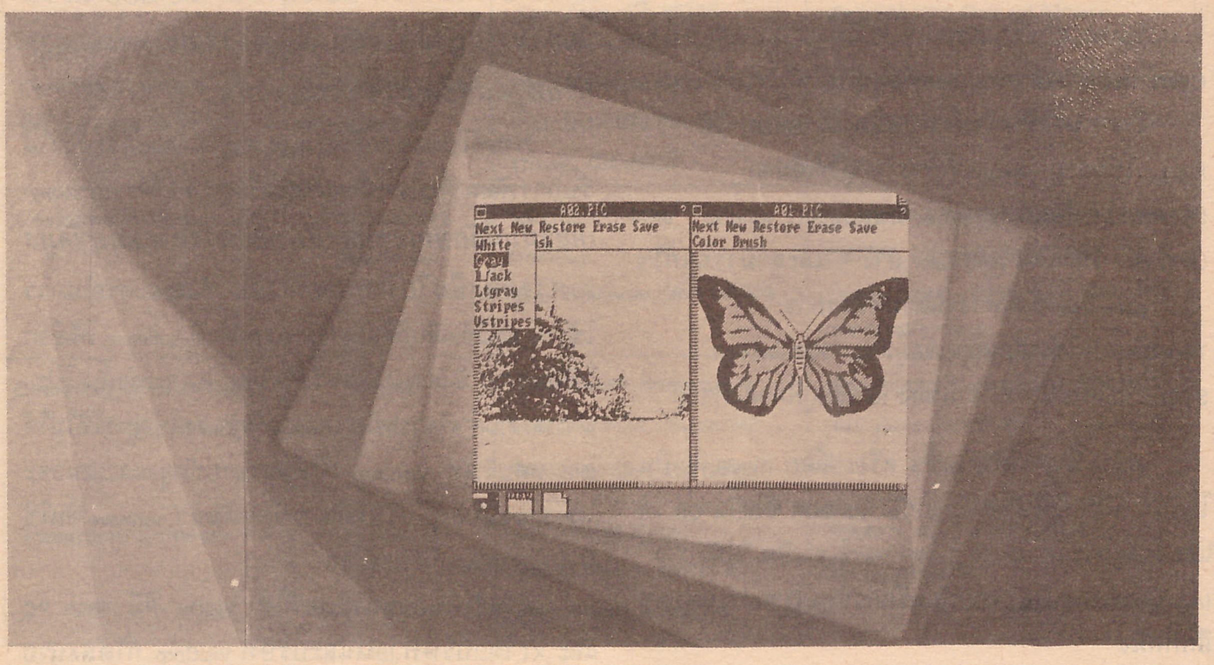


รูปที่ 3, 4 Vision จาก VisiCorp ใช้ window ที่ทับซ้อน กัน

ที่ใช้กับ IBM

นอกจากนี้ยังมีผู้ผลิตทั้ง Hardware และ Software อีกหลายรายที่กำลังจะออกผลิตภัณฑ์ Software และ Hardware ที่สามารถจะใช้กับ VisiOn ได้ตามออกมาในอนาคต ส่วนบริษัท VisiCorp เองก็ได้สร้างโปรแกรมที่ใช้กับ VisiOn ด้วย คือ VisiOnCalc เป็น spreadsheet, VisiOnWord เป็น Word Processor VisiOnGraph เป็นโปรแกรม graphic, VisiOn Query เป็น relational database management system





รูปที่ 5, 6 Microsoft Windows ใช้ window ที่ไม่ซ้อน กัน

ในการใช้ VisiOn นี้จะต้องมี MS-DOS, หน่วยความจำถึง 512 K byte และ Hard disk ขนาด 5 M byte กับ mouse ก็จำเป็นด้วย

Windows

คู่แข่งตัวสำคัญของ VisiOn ก็คือ Microsoft Windows ซึ่งใช้ร่วมกับ MS-DOS เหมือนกัน Microsoft Windows นี้ใช้ระบบ bit-mapped Graphic และใช้ mouse เป็นตัวชี้ตำแหน่งเหมือนกับ VisiOn

ในการทำงานของระบบ Microsoft Windows ได้ขจัดความจำเป็นในการเขียน application software ที่เกี่ยวกับการอ้าง address ทาง hardware โดยตรงด้วยการเพิ่ม function ให้ MS-DOS ในการจัดการกับระบบ bit-mapped screen graphic และระบบ hardware ของ mouse ผลอันนี้จะทำให้โปรแกรมที่เขียนขึ้นมาใช้สำหรับกับ Window ทำงานได้บนเครื่องทุกเครื่องที่ใช้ Operating Environment นี้ไม่ว่าโครงสร้างทาง Hardware จะแตกต่างกันอย่างไร

Microsoft's Windows แตกต่างจาก VisiOn อยู่ 2 ข้อใหญ่ ๆ ข้อแรก คือ *การใช้รูปภาพแสดงแทน function* ในขณะที่ VisiOn ใช้ตัวอักษรแสดงใน windows การเลือก function ทำได้โดยการชี้ไปยังรูปต่าง ๆ ด้วย mouse แล้ว function ที่ถูกชี้นั้นก็จะทำงาน ข้อแตกต่างอีกข้อก็คือวิธีในการแสดงผลบนจอของแต่ละ window *โดย Microsoft Windows จะใช้วิธีการที่เรียกว่า tiling คือวางแต่ละ window แผ่เรียงกันให้เต็มจอ แต่จะไม่มีการซ้อนกัน (Overlapping) ของ window เลยบนจอ* ในการใช้งานปรกติ window ที่ใช้งานจะถูกขยายเต็มบนจอและใน window อื่น ๆ จะถูกลดรูปลงจนกลายเป็นเพียงสัญลักษณ์หรือ icon เล็ก ๆ เท่านั้น ทั้งนี้บริษัท Microsoft เชื่อแน่ว่าเทคนิคดังกล่าวจะทำให้การใช้งานมีประสิทธิภาพมากที่สุด

ผู้ใช้สามารถทำให้ window หลาย ๆ อันปรากฏบนจอได้ในเวลาเดียวกันและขนาดของแต่ละ window ก็สามารถปรับได้ตามความพอใจด้วยคุณสมบัติที่คล้ายการ "Zoom" ซึ่งจะช่วยให้ผู้ใช้แสดงผลได้ในทุกขนาดบนจอ นอกจากนี้ก็สามารถย้าย window กวาดไปทั่วโปรแกรมแบบเดียวกับการที่โปรแกรมหรือ output มีการ scroll ผ่านจอในระบบที่ไม่ใช้ window ได้ จำนวน window ที่จะปรากฏบนจอจะมีได้ไม่จำกัด แต่การใช้งานจริงจะใช้เพียงไม่เกิน 4 window เพราะยิ่งมีหลาย window บนจอก็จะทำให้ข้อมูลภายใน window นั้นเล็กจนอ่านไม่รู้เรื่องทั้งนี้เนื่องจากในการแบ่ง window ของ Microsoft จะไม่มีการทับซ้อนกัน แต่จะแบ่งที่กันไปทั่วจอดังที่กล่าวแล้ว Microsoft Windows นี้สามารถใช้ได้กับโปรแกรมที่เขียนขึ้นจากภาษาใด ๆ ก็ได้ และกับเครื่องใด ๆ ที่ใช้ MS-DOS เป็น OS นอกจากนี้ยังสามารถจะจัดการกับโปรแกรมต่าง ๆ ภายใต้ MS-DOS Version 2.0 ได้ทั้งหมดถึงแม้ว่าโปรแกรมนั้น ๆ จะไม่ได้เขียนขึ้นมาให้ใช้กับระบบ Microsoft Windows ก็ตามโดยในขั้นแรก *ระบบจะสามารถตรวจดูได้ว่าโปรแกรมนั้น ๆ จะใช้ระบบ window ได้หรือไม่ ถ้าไม่ใช้มันก็จะจัดการ save สภาวะการทำงานของ window environment เท่าที่ค้างอยู่เอาไว้ก่อน แล้ว application program นั้นก็จะเริ่มทำงาน เมื่อเสร็จเรียบร้อยระบบ window ก็จะกลับมาทำงานต่อจากที่ค้างอยู่ต่อไป*

Microsoft Windows นี้ไม่มีความต้องการ Hardware ที่ยุ่งยากมากมายเพื่อที่มันจะใช้งานได้ ไม่เหมือนกับ VisiOn โดยมันจะทำงานบนเครื่องคอมพิวเตอร์ที่มี MS-DOS เป็น OS และมีหน่วยความจำ 192 K RAM floppy disk 2 drive และแน่นอนต้องมี mouse ด้วย

CP/M (Concurrent Dos)

บริษัท Digital Research ก็ได้สร้างระบบ window ออกมาด้วยถึงแม้ว่า MS-DOS จะกลายเป็น OS มาตรฐานของระบบ 16 bit ในแทนที่จะเป็น CP/M-86 ของ Digital Research ก็ตาม Digital Research ได้เพิ่มความสามารถให้กับ CP/M จนกลายเป็น Concurrent CP/M โดยทำให้โปรแกรมต่าง ๆ RUN พร้อมกันได้และทำให้เกิดมีระบบ window สำหรับ Concurrent CP/M ขึ้นต่อมา

Concurrent CP/M นั้นสามารถ RUN โปรแกรมต่าง ๆในเวลาเดียวกันได้หลายโปรแกรมหรือที่เรียกว่า Multitasking หรือ Multiprogramming การทำงานมั่นก็คือจะมีการแบ่งส่วนของหน่วยความจำในคอมพิวเตอร์เป็นส่วนย่อย ๆ แยกจากกัน ในแต่ละโปรแกรมก็จะ RUN ภายในหน่วยความจำส่วนของตัวเองไป และ OS จะทำการจัดลำดับการทำงานตามโปรแกรมต่าง ๆ เพื่อจัดการข้อมูลในโปรแกรมนั้น ๆ ในปัจจุบันเมื่อมี window เข้ามาผู้ใช้ *Concurrent CP/M ก็จะเห็นถึงความเป็นไปของโปรแกรมต่าง ๆ ได้พร้อม ๆ กันตลอดเวลาไม่เหมือนกับระบบ window แบบอื่น ๆ ที่มี*

เพียง window เดียวที่ทำงานได้ในขณะหนึ่งแต่ใน Concurrent CP/M นี้ทุกwindow จะทำงานไปพร้อมกัน คืออะไรก็ตามที่เราเห็นใน window เป็นผลของโปรแกรมที่ถูก RUN ในเวลานั้น เราสามารถจะดู file ต่าง ๆ บน disk ในขณะที่ดูการทำงานของโปรแกรมอีกอันหนึ่งพร้อม ๆ กับที่เรากำลังจะส่งข้อมูลจากโปรแกรมหนึ่งไปยังโปรแกรมอื่น ๆ ด้วย ซึ่งเป็นปรากฏการณ์ที่ไม่เคยเกิดมาก่อนในระดับไมโครคอมพิวเตอร์

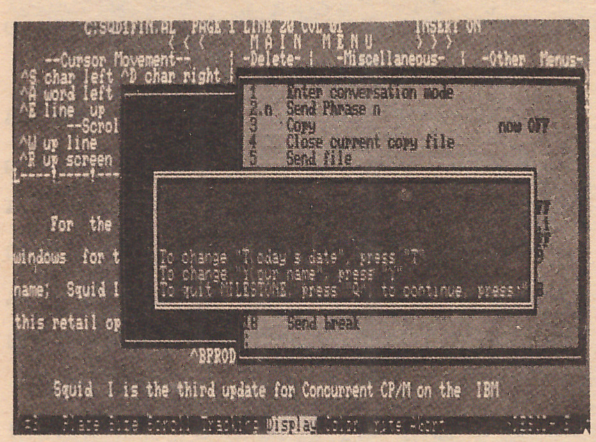
ใน Concurrent CP/M นี้จะใช้พร้อมกันได้มากที่สุด 4 window โดยแต่ละ window ก็จะมีการแยกกันทั้งสีและ Background และจะขยาย, เลื่อนการแสดงผลหรือเลื่อน - window ไปได้ทุกที่หรือแสดงผลเฉพาะ window ที่สำคัญที่สุดก็ได้ข้อมูลจาก file หนึ่งสามารถส่งไปยังโปรแกรมหรือ window อื่นใดก็ได้ ใต้จอจะมี menu line แสดง function ต่าง ๆ ที่ใช้ได้ซึ่งเราเลือกใช้ได้โดยการเลื่อน cursor ไปยังตำแหน่งนั้น ๆ ในการเพิ่ม window เข้าไปในระบบ CP/M นี้ทำให้การจักการกับข้อมูลทำได้ดีขึ้น เช่นการรับข้อมูลจาก data base ระบบหนึ่ง ในขณะที่ทำงานอย่างอื่นไปด้วย

เมื่อเทียบกับระบบ software แบบอื่น ๆ แล้ว Concurrent CP/M เป็น Operating System ที่มีระบบ window ในตัวเลย แทนที่จะเป็นโปรแกรมที่มาดัดแปลง (modify) หรือแทนที่ (overlay) บางส่วนของ Operating System เดิมเหมือนอย่าง software อื่น ๆ นอกจากนี้ Version ใหม่ของ Concurrent CP/M ที่เรียกว่า Concurrent DOS ยังมีส่วนเพิ่มเติมเข้ามาที่เรียกว่า DOS mode คือ mode ที่จะทำให้คุณสามารถ RUN โปรแกรมที่เขียนขึ้นให้ใช้กับ PC-DOS Version 1.1 ได้อีกด้วย โดยรับคำสั่งทั้งของ CP/M เอง และคำสั่งใน PC-DOS

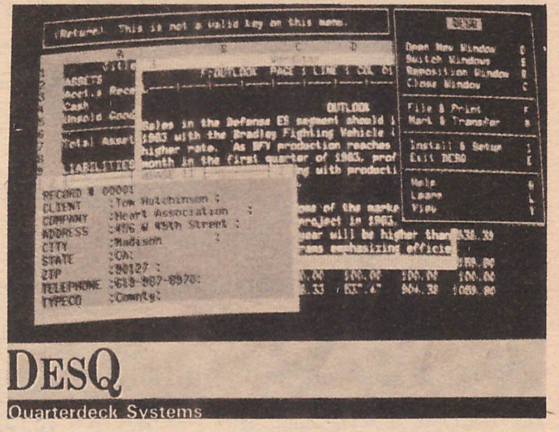
Concurrent CP/M ที่มี window นี้ต้องการหน่วยความจำ 256 k และ floppy disk drive 2 ตัว แต่ไม่จำเป็นต้องใช้ mouse (แต่ผู้ผลิต mouse ก็อาจดัดแปลงให้รับ input จาก mouse ได้) เพราะการเลือก window เลื่อนหรือขยาย window ใช้การกด Control Key ร่วมกับคีย์อื่น ๆ ทำได้

DesQ

ในขณะที่บริษัทต่าง ๆ ที่มีชื่อเสียงเริ่มทำระบบ window ออกมาอย่างเป็นล่ำเป็นสัน บริษัท Quarterdeck Software ก็เป็นบริษัทหนึ่งที่ผลิต Window Software ออกมาชื่อว่า DesQ (อ่านว่าเดสค์ "desk") ระบบ DesQ นี้ผู้ใช้สามารถที่จะใช้งานต่าง ๆ ใน file ได้โดยทันทีเหมือนกับการทำงานกับข้อมูลบน disk โดย file หรือ document จะแสดงบนจอซ้อนกัน (ไม่เหมือนกับ Microsoft's Window) โดย DesQ จะมี menu เป็นตัวอักษรไม่ใช่รูปภาพ ข้อดีของ DesQ คือการใช้งาน application ต่าง ๆ มากหลายชนิดที่ RUN ภายใต้ MS-DOS จะถูกรวมกันภายใต้ DesQ โดยไม่ต้องมีการแก้ไขหรือมีโปรแกรมที่ต้องเขียนใหม่ ผู้ใช้ DesQ สามารถที่จะทำงานอยู่บน file หนึ่งแล้วในขณะเดียวกันอาจจะมีการอ้างถึง window อื่น ๆได้ด้วย DesQ สามารถเอาข้อมูลจากโปรแกรมหนึ่งและใส่เข้าไปยังโปรแกรมอื่น ๆ อยู่คนละ Window ได้ จุดเด่นประการหนึ่งของ DesQ ก็คือ *มีความสามารถในการเรียนรู้การสร้าง routine ในการทำงานซ้ำ ๆ กัน คือผู้ใช้สามารถโปรแกรม DesQ ในการทำงานบาง*



รูปที่ 7 Concurrent CP/M with window จาก Digital Research



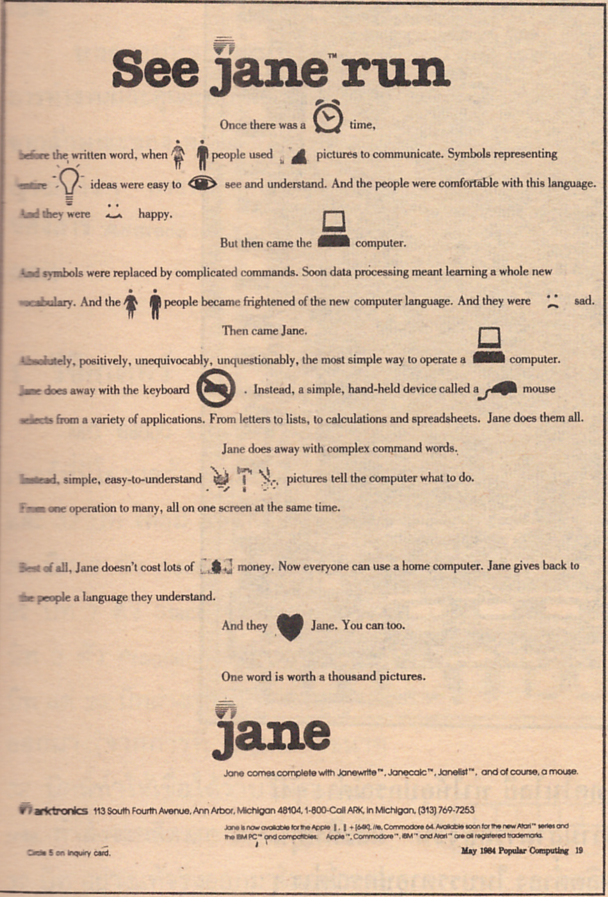
รูปที่ 8 DesQ โต๊ะทำงานอิเลคทรอนิคส์จาก Quaterdeck System

ชนิดที่มีขั้นตอนแน่นอนโดยเขียนเป็นชุดคำสั่งไว้ต่างหากแล้วกำหนดวิธีเรียกใช้ชุดคำสั่งนั้นไว้ เวลาจะใช้ก็ทำได้สะดวกโดยการสั่งเพียงไม่กี่ปุ่มหรือใช้ mouse ช่วย DesQ นี้ถูกออกแบบมาให้ทำงานบน MS-DOS และเครื่องที่มีหน่วยความจำ 512 K RAM และมี Hard disk ขนาด 5 megabyte โดยจะใช้หรือไม่ใช้ mouse ก็ได้

Jane

เครื่องคอมพิวเตอร์ที่ใช้ MS-DOS เป็น OS นั้นไม่ใช่เป็นชนิดเดียวที่มี Window ได้บริษัท Arktronics ซึ่งเป็นบริษัทเล็ก ๆ ได้สร้างโปรแกรมเรียกชื่อว่า Jane ขึ้นโปรแกรมนี้ใช้กับเครื่อง 8 บิท เช่น Commodore 64, Apple ทั้งหลายได้ Jane นี้ไม่ใช่เป็น Software ที่ทำหน้าที่รวมซอฟท์แวร์ตัวอื่น ๆ เหมือนระบบ window อื่น ๆที่กล่าวถึงไปแล้ว แต่เป็น*ชุดของ application program ที่สมบูรณ์ในตัวและใช้ Window เป็นของตัวเอง คือเป็น Integrated Software* นั่นเอง Janeจะประกอบไปด้วย Package 3 ตัวคือ Word Processor, spreadsheet และ file Manager การเลือก Window ที่ปรากฏบนจอและfunction การทำงานต่าง ๆ จะเลือกใช้ได้จาก mouse แต่ก็เป็นเรื่องแน่นอนที่ว่าโปรแกรม Word Processor ของ Jane จะไม่สมบูรณ์แบบและซับซ้อนเหมือนกับโปร แกรม Word Processor ที่เป็น Word Processor โดยเฉพาะ และ package อื่น ๆ ก็เช่นเดียวกัน Jane นี้เปรียบเสมือน Lotus 1-2-3 ที่มาอยู่ในเครื่อง Apple II นั่นเอง นอกจากจะเป็นโปรแกรมสำหรับเครื่อง Appleแล้วก็ยังมีversion ของเครื่อง Commodore 64 (อยู่ในรูป ROM Cartridge) ส่วน version สำหรับเครื่อง Atari และ IBM PC ก็กำลังอยู่ในระหว่างการพัฒนา

เครื่อง Apple ที่ใช้ต้องมีหน่วยความจำขนาด 64 K RAM มี disk อย่างน้อย 1 ตัว และ mouse



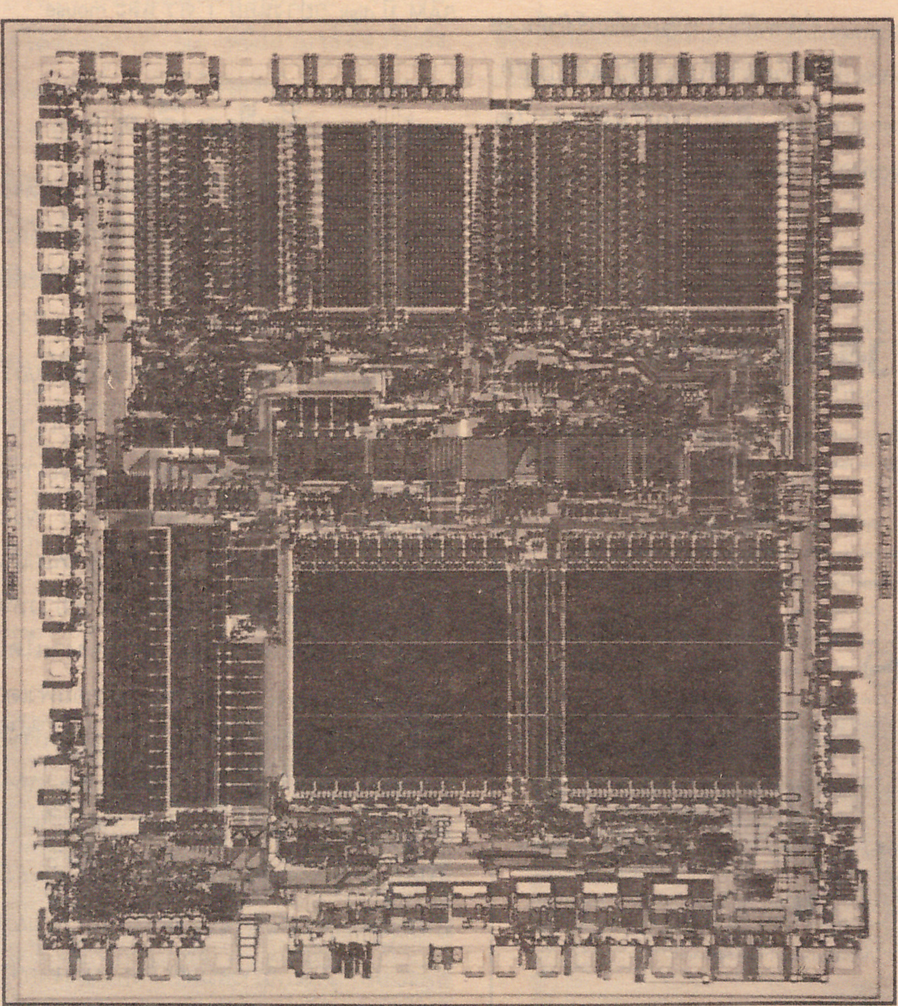
รูปที่ 9 Jane โฆษณาเน้นความง่ายในการใช้งานเป็นสำคัญ

Window อันไหนดี ?

Software ที่กล่าวมาทั้งหมดนั้นเป็น Window Software ที่ออกมาสู่ตลาดในระยะเวลาไล่เลี่ยกัน แต่ละอันก็มีความแตกต่างกัน ทีนี้ก็มาถึงปัญหาที่ว่าเราควรจะเลือกใช้ตัวไหนจึงจะเหมาะ แต่เราตอบคำถามนี้ด้วยถามที่ว่า *เรามีความจำเป็นที่จะใช้ความสามารถในการใช้ข้อมูลร่วมกันในแต่ละโปรแกรมมากน้อยแค่ไหน ?* ถ้าเราจำเป็นต้องใช้ระบบ window ก็เหมาะสมดีแต่ถ้างานของเราทำอยู่แต่เฉพาะ application โปรแกรมตัวใดตัวหนึ่งเช่น Word Processing ละก็การมี window ก็ดูออกจะเป็นการฟุ่มเฟือยเกินไป อย่างไรก็ตามถ้าเราใช้คอมพิวเตอร์ของเราอย่างหนักและมีงานมากมายหลายจำพวกโดยไม่เจาะจงงานใดงานหนึ่งโดยเฉพาะรอบ window ก็จะทำให้เครื่องคอมพิวเตอร์นั้นมีประสิทธิภาพเพิ่มขึ้นอีกมากทีเดียว ยิ่งกว่านั้นถ้าข้อมูลของเราที่เราใช้อยู่ได้รับการ process มาจากโปรแกรมหนึ่งและจำเป็นต้องเอาไปใช้กับอีกโปรแกรมหนึ่งหรือโปรแกรมอื่น ๆ ต่อไปอีกเราก็ควรจะใช้ระบบ window นี้ *ต่อมาเราก็ควรจะคำนึงถึง Hardware ที่จะรองรับ และคุณสมบัติของแต่ละระบบ* ว่าจะเหมาะสมหรือไม่ก็จะทำให้การตัดสินใจของเราถูกต้องยิ่งขึ้นอีกประการหนึ่งก็คือหลังจากที่เราได้เลือก window ที่จะนำมาใช้งานแล้ว เราก็ควรจะดูว่ามี *Software มากน้อยแค่ไหนที่สามารถนำมาใช้กับมันได้* ถ้ามีมากก็ยิ่งทำให้การใช้ application program ร่วมกับ window เกิดประโยชน์มากขึ้น ■

MC 68000 ไมโครโปรเซสเซอร์ 16/32 บิท

โดย ประพันธ์ ธีระวรรณวิไล



บทนำ

นับวันไมโครคอมพิวเตอร์รุ่นใหม่ ๆ ที่ออกมาก็ยิ่งมีประสิทธิภาพสูงขึ้นเรื่อย ๆ ในราคาที่ถูกลง ทั้งนี้เพราะความก้าวหน้าทางเทคโนโลยี ทำให้อุปกรณ์ต่าง ๆ มีขีดความสามารถสูงขึ้น และต้นทุนการผลิตต่ำลง ในบรรดาอุปกรณ์ต่าง ๆ ที่ประกอบกันเป็นไมโครคอมพิวเตอร์

MC 68000

นั้น CPU หรือไมโครโปรเซสเซอร์นั้นนับเป็นหัวใจสำคัญและมีบทบาทมากที่สุด ที่จะกำหนดว่า ไมโครคอมพิวเตอร์เครื่องนั้น ๆ จะทำอะไรได้มากแค่ไหน

ไมโครโปรเซสเซอร์เบอร์หนึ่งที่กำลังได้รับความนิยมมากก็คือ MC 68000 ของบริษัทโมโตโรลา เครื่องรุ่นใหม่ ๆที่ใช้ MC 68000 (ซึ่งบางทีก็เรียกว่า 68000 เฉย ๆ หรือเรียกย่อเป็น 68 K หรือ 1 K = 1000 ดังนั้น 68K ก็คือ 68×1000 = 68000 นั่นเอง) ก็มีอาทิเช่น Macintosh ของ Apple,Sinclair OL (ใช้ 68008 ซึ่งอยู่ในตระกูลเดียวกับ 68000) ตลอดจน IBM XT/370 Pcrsonal Computer รุ่นใหม่ของ IBM

บทความนี้จะกล่าวถึงลักษณะการทำงานและคุณสมบัติต่าง ๆ โดยเริ่มจากลักษณะของ Register ภายใน สัญญาณต่าง ๆ , การทำ Interrupt, ชนิดของการ Address และชุดคำสั่งของ MC 68000

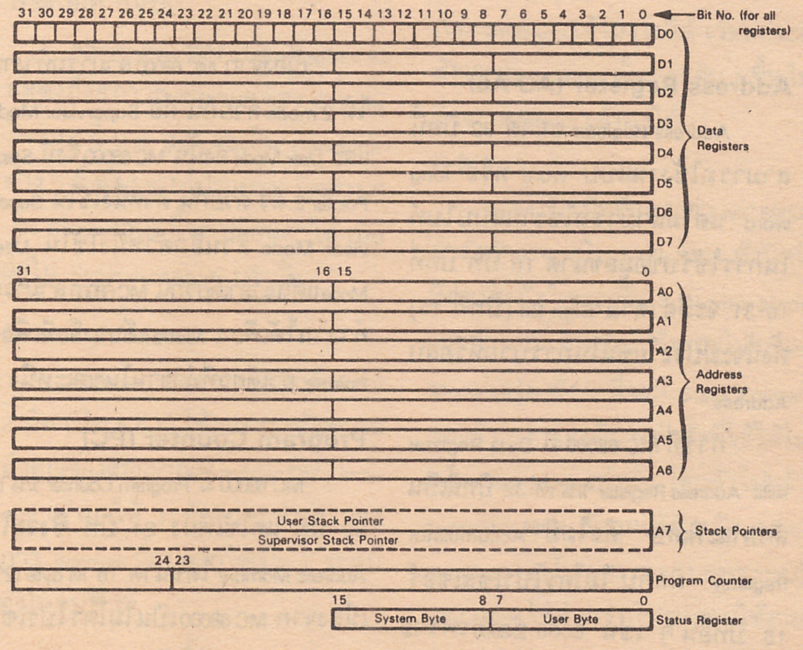
1. รีจิสเตอร์

MC 68000 เป็นไมโครโปรเซสเซอร์ที่มีโครงสร้างทั้งแบบ 16 บิท และ 32 บิท โดยมี Data Register ขนาด 32 บิท 8 ตัว Address Register ขนาด 32 บิท 7 ตัว Stack Pointer ขนาด 32 บิท 2 ตัว Program Counter ขนาด 32 บิท แต่ 24 บิทล่างเท่านั้นที่ส่งสัญญาณออกมา (จากการที่ Register มีขนาด 32 บิทนี้ทำให้เชื่อได้ว่าบริษัท Motorola กำลังจะผลิต MC 68000 รุ่นที่เป็น 32 บิท อย่างแท้จริงออกมาในอนาคต) และ Status Register ขนาด 16 บิท รีจิสเตอร์ต่าง ๆ ได้แสดงไว้ในรูปด้านล่าง ๆ

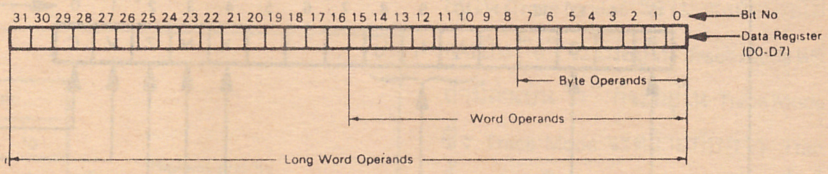
Data Register (Do-D7)

MC 68000 มี Data Register ขนาด 32 บิท แต่สามารถใช้งานแบบไบท์ (8 บิท) word (16 บิท) หรือ long word (32 บิท) ก็ได้ ตำแหน่งบิทต่าง ๆ ที่ใช้งานได้แสดงไว้ในรูปที่ 2

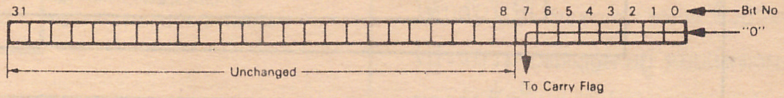
ในการใช้งาน Data Register ไม่ถึง 32 บิท บิทที่ไม่เกี่ยวข้องจะไม่มีการเปลี่ยนแปลงดังในรูปที่ 3 ซึ่งแสดงถึงการ Shift แบบ Arithmatic ขนาด 8 บิท ขอให้สังเกตว่า บิทที่ 8-31 ไม่มีการ



รูปที่ 1 Programmable Register ใน 68000



รูปที่ 2



รูปที่ 3

เปลี่ยนแปลง หรือเกี่ยวข้องด้วยเลย

Data Register นี้นอกจากจะใช้เป็นตัวรับส่งข้อมูลแล้ว ยังสามารถใช้เป็น Index Register ได้ (Index Register คือ รีจิสเตอร์ที่ค่าภายในจะถูกนำไปบวกในการคำนวณหา Address ซึ่งจะกล่าวใน Addressing Mode)

Address Register (AO-A6)

Address Register ขนาด 32 บิทนี้สามารถใช้งานแบบ word หรือ long word แต่ไม่สามารถใช้งานแบบไบท์ ในการใช้รับข้อมูลขนาด 16 บิท บิทที่ 16-31 จะมีค่าตาม Sign Bit (บิทที่ 15) ทั้งนี้จะมีประโยชน์ในการบวกหรือลบ Address

การที่ MC 68000 มี Data Register และ Address Register ขนาด 32 บิทนี้เป็นลักษณะพิเศษ คือไม่มี Accumulator Register ผิดกับ ไมโครโปรเซสเซอร์ 16 บิทอื่น ๆ เช่น 8086 Z-8000 เพราะ MC 68000 สามารถใช้รีจิสเตอร์ตัวไหนเป็น Accumulator ก็ได้ ไม่มีการเจาะจงรีจิสเตอร์ตัวหนึ่งตัวใดโดยเฉพาะ วิธีการนี้ใช้ในเครื่องมินิคอมพิวเตอร์ หรือเครื่องเมนเฟรม อย่างเช่นเครื่อง IBM 370

Stack Pointer (A7)

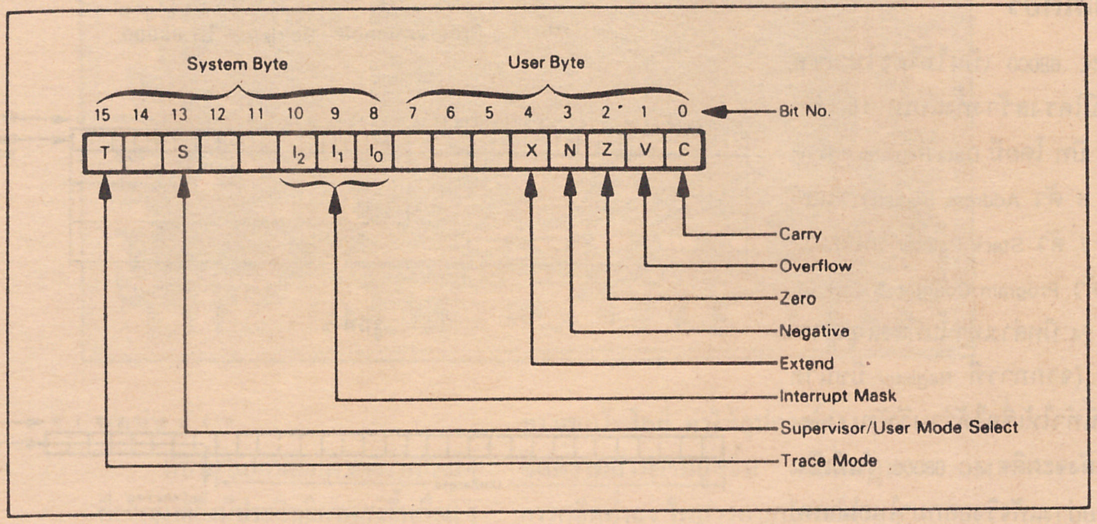
เนื่องจาก MC 68000 สามารถทำงานได้ 2 mode ด้วยกัน คือ Supervisor Mode และ User Mode ดังนั้น MC 68000 จึงมี Stack Pointer 2 ตัว ด้วยกัน ตัวหนึ่งใช้ใน Supervisor Mode ส่วนอีกตัวหนึ่งใช้ใน User Mode แต่ในเวลาเดียวกัน MC 68000 สามารถทำงานได้เพียง Mode เดียว จึงมีเพียง Pointer ตัวเดียวที่ทำงานในขณะหนึ่ง ๆ

Program Counter (PC)

MC 68000 มี Program Counter ขนาด 32 บิท แต่ใช้เพียง 24 บิท ซึ่งจะใช้ Address Momory ได้ขนาด 16 M byte (2^{24}) เนื่องจาก MC 68000 เป็นไมโครโปรเซสเซอร์ขนาด 16 บิท ดังนั้น ข้อมูลพื้นฐานจะมีขนาด 16 บิท (word) แต่สามารถเรียกใช้แบบ 8 บิท หรือ 32 บิทก็ได้ ในการเรียกใช้ข้อมูลขนาด 16 บิท (word) หรือ 32 บิท (long word) นี้ Address จะต้องเป็นเลขคู่ ลักษณะเช่นนี้เป็นลักษณะทั่วไปของไมโครโปรเซสเซอร์ขนาด 16 บิท ซึ่งมีผลทำให้ค่าใน Program Counter และ Stack Pointer ต้องเป็นเลขคู่ (บิท 0 มีค่าเป็น "0") ตามไปด้วย MC 68000 จะมีคุณสมบัติพิเศษอีกข้อหนึ่งคือจะคอยเช็คว่าการเรียกใช้ข้อมูลขนาด 16 บิทหรือ 32 บิท มี Address เป็นเลขคู่หรือไม่ ถ้าไม่ก็จะเกิดการ Exception ขึ้น (Exception คือการเกิด Interrupt นั่นเอง ซึ่งจะกล่าวถึงต่อไปในหัวข้อ Exception)

Status Register

MC 68000 ใช้ Status Register ขนาด 16 บิท ซึ่งแบ่งเป็น 2 ส่วน คือ System Byte และ User Byte รูปด้านล่างจะแสดง



รูปที่ 4 บิทต่าง ๆ ใน Statur Register

ถึงหน้าที่ของแต่ละบิทใน Status Register

User Byte

- บิท Carry (C) ทำหน้าที่เหมือนกับ Carry ในไมโครโปรเซสเซอร์ขนาด 8 บิท ทั่ว ๆ ไป คือจะเป็น "1" เมื่อมีตัวทดจากการบวก หรือมีตัวยืมจากการลบ นอกจากนี้แล้วคำสั่ง Rotate และ Shift ยังมีผลกับบิท ๆ นี้

- บิท Overflow (V) บิทนี้ใช้ในการคำนวณเลขที่มีเครื่องหมาย ค่าที่ได้เกิดจากการ Exclusive OR ระหว่าง Carry Flag กับ Sign Bit การที่บิทนี้มีค่าเป็น "1" แสดงว่าขนาดคำตอบใหญ่เกินกว่าที่รีจิสเตอร์จะเก็บค่าได้

- บิท Zero (Z) บิดนี้จะมีค่าเป็น "1" เมื่อคำตอบมีค่าเป็น 0 เท่านั้น นอกจากนี้แล้วบิทนี้จะมีค่าเป็น "0"

- บิท Negative (N) บิทนี้จะมีค่าตาม sign บิทของคำตอบ ถ้าบิทนี้มีค่าเป็น "0" แสดงว่าคำตอบเป็นเลขบวก แต่ถ้าบิทนี้มีค่าเป็น "1" แสดงว่าคำตอบเป็นเลขลบ

- บิท Extend (X) บิทนี้ใช้เป็น Carry ในการคำนวณที่ต้องการความเที่ยงตรงมาก ๆ (Multipresision) คำสั่งที่มีผลกับบิทนี้อย่างเช่นคำสั่งบวก คำสั่งลบ คำสั่ง Shift บิทนี้จะมีค่าตามบิท Carry

บิทที่เหลืออยู่ ในปัจจุบันยังไม่มีการใช้งาน และมีค่าเป็น "0" เสมอ

System Byte

System Byte ของ Status Register จะแสดงถึงสภาวะ (Status) ของ System ในขณะที่ User Byte จะเป็น Condition ต่าง ๆ ซึ่งมีผลจากโปรแกรม บิทต่าง ๆ ใน System Byte นี้จะเปลี่ยนค่าได้ เมื่ออยู่ใน Supervisor Mode เท่านั้น

Interrupt Mask

Interrupt Mask ประกอบด้วยบิท 3 บิท (บิท 5,6,7) ใช้แสดงระดับความสำคัญของสัญญาณ Interrupt ที่จะ Interrupt เข้ามาได้ ระดับความสำคัญของการ Interrupt นี้แบ่งออกได้เป็น 7 ระดับ ดังในรูปที่ 5

จากรูปจะเห็นได้ว่า Interrupt ระดับ 7 มีระดับความสำคัญสูงสุดซึ่งจะ Interrupt เข้ามาได้ตลอดเวลา User ไม่สามารถทำการป้องกันการ Interrupt หรือทำการ Mask ไว้ได้จึงเป็น Non-maskable interrupt ระดับความสำคัญของการ Interrupt นี้มีชื่อว่า Priority โดย Priority ระดับ 7 เป็น Priority สูงสุดคราวนี้สมมติว่าเรา Mask ไว้ให้มี Priority ระดับ 5 ถ้าเกิดสัญญาณ Interrupt ระดับ 3 **ขึ้น** ก็จะ Interrupt เข้ามาไม่ได้ เว้นแต่จะมีสัญญาณ Interrupt ตั้งแต่ระดับ 5 **ขึ้น**ไปถึงจะ Inrerrupt เข้ามาได้

S-bit

S-bit เป็นบิทที่แสดงว่า MC 68000 กำลังทำงานอยู่ใน Mode ไหน ถ้าบิทนี้มีค่าเป็น "1" จะอยู่ใน Supervisor Mode และถ้าบิทนี้เป็น "0" ก็จะอยู่ใน User Mode Mode ทั้ง 2 นี้จะมี Stack Pointer ของตัวเอง และมีบางคำสั่งที่ใช้ได้ใน Supervisor Mode เท่านั้น โดยทั่วไปแล้ว Supervisor Mode มักถูกใช้สำหรับ Operating System (โปรแกรมควบคุมระบบ) User Mode ใช้สำหรับโปรแกรมใช้งานต่าง ๆ (Application Program)

I-bit

I-Bit นี้ใช้แสดงว่า ขณะนั้น MC 68000 อยู่ใน Trace Mode หรือไม่ ซึ่งถ้าบิทนี้มีค่าเป็น "0" จะไม่อยู่ใน Trace Mode แต่ถ้ามีค่าเป็น "1" ก็จะอยู่ใน Trace Mode ซึ่ง Trace Mode จะคล้ายกับการ Trap ขึ้น (Trap คือการ Interrupt ชนิดหนึ่งที่เกิดโดย Software ไม่ใช่สัญญาณ Interrupt สั่งเข้ามาจากอุปกรณ์ภายนอก) และทำให้กลับไปที่ Supervisor Mode ซึ่งอาจมีโปรแกรม Debugger อยู่ คอยแสดงผลของคำสั่งนั้น ๆ (โปรแกรม Debugger เป็นโปรแกรมที่ใช้ดูว่าโปรแกรมอื่นทำ

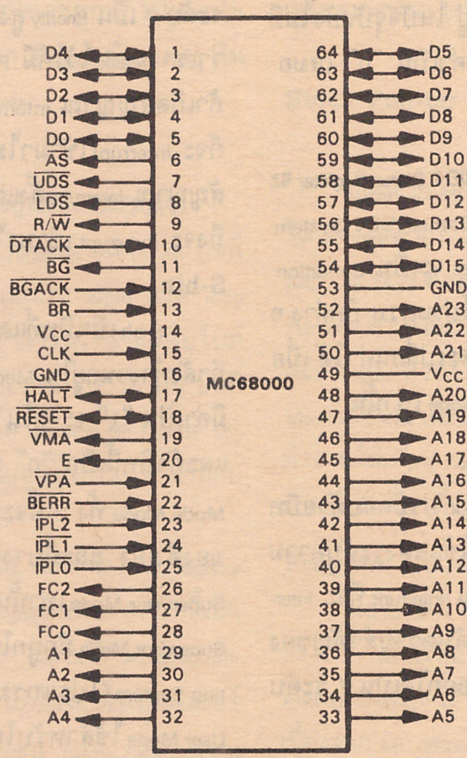
Interrupt Level	Interrupt Mask			
	I_2	I_1	I_0	
Level 7	1	1	1	← Highest priority (Non-maskable)
Level 6	1	1	0	
Level 5	1	0	1	
Level 4	1	0	0	
Level 3	0	1	1	
Level 2	0	1	0	
Level 1	0	0	1	← Lowest Priority
Level 0	0	0	0	← No Interrupt Request

รูปที่ 5

อะไรบ้าง ตัวอย่างของโปรแกรมชนิดนี้ คือโปรแกรม DDT หรือ Dynamic Bebuging Tool ใน CP/M)

2. ขาและสัญญาณควบคุม

MC 68000 เป็นไอซีขนาด 64 ขา ดังในรูป ขอให้สังเกตว่า ที่ขาแต่ละขาจะมีชื่อย่อของสัญญาณ และบางตัวยังมีขีดอยู่ข้างบน เช่น $\overline{UDS}$ ทั้งนี้เพื่อที่จะแยกได้ว่าสัญญาณไหนทำงานที่ลอจิกอะไร ถ้าชื่อย่อของสัญญาณไม่มีขีดถือว่า ทำงานเมื่อสัญญาณเข้ามีลอจิก "1" (Active High) แต่ถ้ามีขีด จะทำงานเมื่อมีลอจิก "0" (Active Low)



Pin Name	Description	Type
D0-D15	Data Bus	Bidirectional, Tristate
A1-A23	Address Bus	Output, Tristate
$\overline{AS}$	Address Strobe	Output, Tristate
R/$\overline{W}$	Read/Write Control	Output, Tristate
$\overline{UDS}$, $\overline{LDS}$	Upper, Lower Data Strobes	Output, Tristate
$\overline{DTACK}$	Data Transfer Acknowledge	Input
FC0, FC1, FC2	Function Code (status) Outputs	Output, Tristate
$\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$	Interrupt Requests	Input
$\overline{BERR}$	Bus Error	Input
$\overline{HALT}$	Halt Processor Operation	Input/Output
$\overline{RESET}$	Reset Processor or Reset External Devices	Input/Output
CLK	System Clock	Input
$\overline{BR}$	Bus Request	Input
$\overline{BG}$	Bus Grant	Output
$\overline{BGACK}$	Bus Grant Acknowledge	Input
E	Enable (Clock) Output	Output
$\overline{VMA}$	Valid Memory Address	Output, Tristate
$\overline{VPA}$	Valid Peripheral Address	Input
V_{CC}, GND	Power (+5 V) and Ground	

รูปที่ 6 ขาและสัญญาณควบคุมของ 68000

Data Bus และ address Bus

เนื่องจาก MC 86000 เป็นไมโครโปรเซสเซอร์ขนาด 16 บิท ดังนั้นหน่วยของข้อมูลที่ใช้ทั่วไปจะมีขนาด 16 บิท ซึ่งหมายความว่า ในการรับส่งข้อมูลแต่ละครั้ง จะรับส่งได้ไม่เกิน 16 บิท ถ้าจะรับส่งเกิน 16 บิท ก็ต้องรับส่งมากกว่า 1 ครั้ง ข้อมูลที่รับส่งระหว่าง MC 68000 กับอุปกรณ์ภายนอก (อาจจะเป็น Memory หรืออุปกรณ์อินพุท เอาท์พุท ต่าง ๆ) จะรับส่งผ่าน Data Bus (Do-D 15)

เพื่อที่จะให้อุปกรณ์ภายนอกทราบว่า MC 68000 จะติดต่อกับอุปกรณ์ไหน MC 68000 จะใช้สัญญาณจาก Address Bus (A1-A23) เลือกอุปกรณ์โดยส่ง Address ของอุปกรณ์นั้น ไปตามขา A1-A23 พร้อมกับมีสัญญาณ Address Strobe ($\overline{AS}$) เพื่อบอกอุปกรณ์ภายนอกว่าตอนนี้ Address บน Address Bus เป็น Address ที่ถูกต้องแล้ว

MC 68000 ใช้หลักการของ Memory Mopped I/O คือ จะถือว่าอุปกรณ์อินพุทและเอาท์พุท เป็นเสมือนหน่วยความจำ ตำแหน่งหนึ่งที่จะอ่านหรือเขียนข้อมูล ได้เช่น

MC 68000

เกี่ยวกับหน่วยความจำทั่วไป ใน ระบบของไมโครโปรเซสเซอร์ขนาด 8 บิททั่วไปนั้น การรับส่งข้อมูลจะใช้แบบ Synchronous ซึ่งหมายความว่าจะต้องรับส่งข้อมูลให้เสร็จภายในเวลาที่กำหนด ถ้าจะรับส่งกับอุปกรณ์ที่ช้าจำเป็นจะต้องมีต่อวงจรพิเศษเพิ่มเติม หรือใช้ Software ช่วย แต่ MC 68000 นอกจากจะรับส่งแบบ Synchronous แล้ว ยังสามารถรับส่งแบบ Asynchronous ได้อีกด้วย

การรับส่งข้อมูลแบบAsynchronous

อย่างที่เขียนไว้ในตอนต้น MC 68000 สามารถเรียกใช้ข้อมูลขนาด 1 ไบท์ได้ ทำให้ MC 68000 สามารถที่จะอ้าง Address ได้ถึง 16M ซึ่งต้องใช้สัญญาณ Address ขนาด 24 บิท แต่ MC 68000 มีขา Address เพียง 23 ขา ดังนั้นจึงมีสัญญาณ $\overline{LDS}$ (Lower Data Strobe) และ $\overline{UDS}$ (Upper Data Strobe) ช่วย โดยเมื่อสัญญาณ $\overline{UDS}$ เป็น "0" ข้อมูลจะถูกรับส่งใน Data Bus 8 บิทล่าง (D0-D7) และถ้าสัญญาณ $\overline{UDS}$ เป็น "0" ข้อมูลจะถูกรับส่งใน Data Bus 8 บิทบน (D0-D15) ดังนั้นถ้าสัญญาณทั้งสองเป็น "0" ข้อมูลทั้ง 16 บิทจะถูกรับส่งพร้อม ๆ กัน

ต่อมาเพื่อที่จะให้อุปกรณ์ภายนอกทราบว่า MC 68000 กำลังจะรับ (Read) หรือส่ง (Write) MC 68000 จะใช้สัญญาณ R/$\overline{W}$ โดยเมื่อสัญญาณ R/$\overline{W}$ เป็น "1" แสดงว่า MC 68000 กำลังต้องการรับข้อมูล และถ้าเป็น "0" แสดงว่ากำลังส่งข้อมูล การทำงานของสัญญาณ $\overline{UDS}$, $\overline{LDS}$ และ R/$\overline{W}$ ได้สรุปไว้ในตารางด้านล่าง

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8-D15	D0-D7	Operation
High	High				
Low	Low	High	Data bits 8-15	Data bits 0-7	Word Read
High	Low	High		Data bits 0-7	Byte Read
Low	High	High	Data bits 8-15		Byte Read
Low	Low	Low	Data bits 8-15	Data bits 0-7	Word Write
High	Low	Low	Data bits 0-7	Data bits 0-7	Byte Write
Low	High	Low	Data bits 8-15	Data bits 8-15	Byte Write

☐ No valid data output or input

รูปที่ 7 การทำงานของสัญญาณควบคุม BUS ($\overline{UDS}$, $\overline{LDS}$ และ R/$\overline{W}$)

หลังจากที่เราสามารถบอกได้ว่าอุปกรณ์ไหนที่ใช้รับส่ง ข้อมูลไบท์ไหน หรือทั้ง 2 ไบท์ และจะทำการรับหรือส่งข้อมูลแล้ว สิ่งต่อไปก็คือการตอบรับของอุปกรณ์ภายนอกมายัง MC 68000 ว่าส่งข้อมูลมาให้แล้ว หรือรับข้อมูลไว้แล้ว ซึ่งจะใช้สัญญาณ Data Transfer Acknowledge ($\overline{DTACK}$) ถ้าไม่มีสัญญาณนี้ตอบรับกลับมา MC 68000 จะรอไปเรื่อย ๆ วิธีนี้จะทำให้ MC 68000 สามารถที่จะชลอการทำงาน เมื่อติดต่อกับอุปกรณ์ที่ทำงานช้า และทำงานได้เร็วขึ้น เมื่อติดต่อกับอุปกรณ์ที่ทำงานได้เร็ว จะเห็นได้ว่า การรับส่งข้อมูลแบบนี้จะอาศัยการตอบรับ (hand shaking) ไม่ใช่กำหนดช่วงเวลาให้รับส่งให้ทัน การรับส่งข้อมูลโดยอาศัยการตอบรับนี้การรับส่งเป็นแบบ Asynchronous

สัญญาณ Function Code

MC 68000 นอกจากจะมีสัญญาณตอบรับสำหรับส่งข้อมูลแล้ว ยังมีสัญญาณที่ใช้บอกเหตุการณ์ภายนอกว่าข้อมูลที่รับส่งนั้นเป็นข้อมูลอะไรเช่น โปรแกรมของ User หรือข้อมูลของ User หรือโปรแกรมของ Supervisor หรือข้อมูลของ Supervisor สัญญาณที่ใช้บอกนี้คือสัญญาณ Function Code (FCO, FC1, FC2) ความหมายของสัญญาณเหล่านี้ได้แสดงไว้ในรูปที่ 8

สัญญาณ Function Code นี้ สามารถใช้ขยายเนื้อที่ Memory จาก 16 M byte เป็น 64M Byte ได้ นอกจากนี้แล้วยังสามารถใช้สัญญาณ Function Code จัดการเกี่ยวกับ Memory อย่างเช่นไม่ให้ User Program ใช้เนื้อที่ Memory บางส่วนที่สงวนไว้สำหรับการทำงานใน Supervisor mode

การรับส่งข้อมูลแบบ Synchronous

นอกจากจะรับส่งข้อมูลแบบ Asynchronous ซึ่งใช้การตอบรับโดยไม่ขึ้นกับความเร็วของอุปกรณ์ที่ติดต่อด้วยแล้ว MC 68000 ยังสามารถรับส่งข้อมูลแบบ Synchronous การรับส่งแบบ Synchronous นี้ MC68000 ไม่ได้รอ

Table 2-3. Function Code Summary

FC2	FC1	FC0	ชนิดของการรับส่ง
0	0	0	
0	0	1	การรับส่งข้อมูลของ User
0	1	0	การรับส่งโปรแกรมของ User
0	1	1	
1	0	0	
1	0	1	การรับส่งข้อมูลของ Supervisor
1	1	0	การรับส่งโปรแกรมของ Supervisor
1	1	1	ตอบรับการ interrupt

☐ ปัจจุบันยังไม่ได้ใช้งาน

ปที่ 8 การทำงานของสัญญาณ Funciton code (FCO, FC1 และ FC2)

สัญญาณ $\overline{\text{DTACK}}$ แต่ใช้สัญญาณ Clock เพื่อกำหนดช่วงเวลาในการรับส่งแทน

การรับส่งแบบ Synchronous ใช้สัญญาณ Enable (E) เป็นตัวส่งสัญญาณ Clock ดังกล่าว จาก MC 68000 โดยใช้ความถี่เพียง 1/10 ของสัญญาณ Clock ที่ MC 68000 ใช้จริง และรูปสัญญาณจะมีขนาด 40/60 คือเป็น "1" อยู่ 4 Clock (ของ MC 68000) และเป็น "0" อยู่ 6 Clock

การรับส่งแบบ Synchronous นี้ อุปกรณ์ภายนอกจะส่งสัญญาณ Valid Peripheral Acknowledge ($\overline{\text{VPA}}$) มาให้ MC 68000 เพื่อบอกให้เราทราบว่าต่อไปจะรับส่งข้อมูลแบบ Synchronous MC 68000 จะตอบว่ารับทราบแล้วโดยใช้สัญญาณ Valid Memory Achnowledge (VMA)

การที่ MC 68000 สามารถที่จะรับส่งข้อมูลแบบ Synchronous ทำให้ MC 68000 สามารถใช้อุปกรณ์สนับสนุนที่เคยใช้กับ MC 68000 ได้ MC 6800 เป็นไมโครโปรเซสเซอร์ขนาด 8 บิท ของ Motorola เช่นเดียวกันและจัดว่าเป็นไมโครโปรเซสเซอร์ตระกูลหนึ่งที่มีชื่อเสียงและมีอุปกรณ์ประกอบเป็นจำนวนมาก ซึ่งอุปกรณ์เหล่านั้นส่วนมากก็มีการรับส่งข้อมูลแบบ Synchronous ด้วยวิธีนี้ MC 68000 ก็จะมีอุปกรณ์ประกอบเพิ่มขึ้นอีกมาก ซึ่งไมโครโปรเซสเซอร์ขนาด 16 บิท อื่น ๆ ก็ใช้วิธีนี้อย่างเช่น 8086 ของ Intel สามารถใช้อุปกรณ์ร่วมกับ 8080 อันเป็นไมโครโปรเซสเซอร์ 8 บิท ที่มีชื่อเสียงของบริษัท Intel เองได้

สัญญาณควบคุม Bus และระบบ

สัญญาณ Bus Request ($\overline{\text{BR}}$) Bus Grant ($\overline{\text{BG}}$) และ Bus Grant Acknowledge ($\overline{\text{BGACK}}$) เป็นสัญญาณควบคุม Bus สำหรับการทำ DMA (Direct Memory Access) สัญญาณ $\overline{\text{RESET}}$, Bus Error ($\overline{\text{BERR}}$) และ $\overline{\text{HALT}}$ เป็นสัญญาณ 2 ทิศทาง นอกจากอุปกรณ์ภายนอกจะส่งสัญญาณเข้ามาแล้ว MC 68000 ยังสามารถส่งสัญญาณนี้ไปควบคุมอุปกรณ์ภายนอกโดยตัว MC 68000 ไม่ต้องถูก Halt หรือ Reset จริง ๆ อีกด้วย สัญญาณ BERR เป็นสัญญาณที่ใช้ประกอบกับ $\overline{\text{DTACK}}$ โดยเมื่ออุปกรณ์ภายนอกเกิดความผิดพลาด ไม่สามารถที่จะส่งสัญญาณ $\overline{\text{DTACK}}$ ได้สัญญาณ $\overline{\text{BERR}}$ ก็จะถูกส่งเข้ามาบอก MC 68000 (อาจจะส่งมาจากอุปกรณ์อีกตัวหนึ่ง ซึ่งทำหน้าที่ตรวจสอบความผิดพลาด) ว่ามี Error เกิดขึ้น

สัญญาณ Interrupt

สัญญาณ Interrupt ประกอบด้วยสัญญาณ 3 สัญญาณ คือ สัญญาณ $\overline{\text{IPO}}$, $\overline{\text{IP1}}$ และ $\overline{\text{IP2}}$ ซึ่งรวมกันแล้ว จะแสดงถึงระดับ Priority หรือระดับความสำคัญของการ Interrupt การที่สัญญาณ Interrupt จะ Interrupt เข้ามาได้นั้นจะต้องมีระดับ Priority สูงกว่า หรือเท่ากับที่ Mask ไว้ใน Status Register ถ้าสามารถ Interrupt เข้ามาได้ MC 68000 ก็จะไปทำ Routine ต่าง ๆ ตาม Vector ซึ่งกล่าวต่อไปในหัวข้อ Exception (Vector คือ Address ของ Routine ต่าง ๆ)

3. Exception

Exception คือการ Interrupt แต่บริษัท Motorola ใช้ชื่อนี้ก็เพราะ MC 68000 สามารถทำ Interrupt ได้มากชนิดกว่าไมโครโปรเซสเซอร์ชนิดอื่นมาก

วิธีการจัดการกับ Exception นั้น MC 68000 จะใช้ตารางซึ่งเก็บ Vector หรือ Address ของ Routine ที่จะจัดการ

การ Interrupt ชนิดต่าง ๆ เมื่อ MC 68000 รู้ว่าเป็น Interrupt ชนิดไหนก็จะสามารถไปทำ Routine นั้น ๆ ตาม Vector ในตาราง ตารางที่ใช้เก็บ Vector นี้มีชื่อว่า Exception Vector Table ไมโครโปรเซสเซอร์ขนาด 16 บิท ชนิดอื่น เช่น 8086 และ 8000 ก็ใช้วิธีเดียวกันนี้ ต่างกันที่ว่า 68000 มีตารางที่ใหญ่กว่ามาก นอกจากนี้ MC 68000 ยังสามารถแบ่งการ Interrupt จากภายนอกที่เข้ามาได้ถึง 7 ระดับ

Mode ของการทำงานกับการเกิด Exception

อย่างที่ได้กล่าวไว้ในตอนต้น MC 68000 สามารถทำงานได้ 2 Mode คือ Supervisor Mode และ User Mode เมื่อ MC 68000 ถูก Reset มันจะเริ่มทำงานใน Supervisor Mode จนกว่าจะมีคำสั่งที่เปลี่ยนค่า Bit ใน Status Register ให้เป็น "0" MC 68000 ก็จะไปทำงานใน User Mode ต่อไป

เมื่อ MC 68000 ทำงานใน User Mode มีแต่การ Exception เท่านั้นการที่จะทำให้ MC 68000 กลับไปทำงานใน Supervisor Mode ได้ การ Exception ทุกชนิดจะทำให้ MC 68000 กลับทำงานใน Supervisor Mode ทั้งสิ้น

ชนิดของ Exception

Exception แบ่งออกได้เป็น 2 ชนิดใหญ่ ๆ ด้วยกัน คือ

1) Exception ที่เกิดจากภายใน ซึ่งมีผลเกิดจากคำสั่งบางคำสั่งหรือเกิดความผิดพลาดบางอย่าง

2) Exception ที่เกิดจากภายนอก ได้แก่สัญญาณที่อุปกรณ์ภายนอกส่งเข้ามา Interrupt

Exception ที่เกิดขึ้นภายใน

Exception ที่เกิดจากภายใน มีสาเหตุที่เกิดขึ้นได้ 3 อย่าง คือ ภายในพบความผิดพลาด, ทำคำสั่งจำพวก Trap (Trap เป็นคำสั่งที่มีผลให้เกิด Exception เรามักจะเรียกคำสั่งพวกนี้ว่า Software Interrupt ซึ่งเทียบได้กับคำสั่ง Restart ของ 8080) และฟังก์ชั่น Trap ความผิดพลาดที่พบภายใน ได้แก่

1) Address ผิด (Address Error) MC 68000 พยายามที่จะเรียกใช้ข้อมูลขนาด 16 บิท หรือ 32 บิท หรือคำสั่งที่มี Address เป็นเลขคี่

2) พยายามที่จะทำคำสั่งที่ไม่อนุญาตให้ใช้ใน User Mode

3) คำสั่งผิดหรือไม่มีในชุดคำสั่ง MC 68000 นอกจากจะมีคำสั่ง Trap โดยตรงแล้ว ยังมีคำสั่งอื่นอีก ที่จะทำให้เกิดการ Trap เมื่อมีเงื่อนไขสอดคล้องกับที่ต้องการ อย่างเช่น ให้ Trap เมื่อเกิด Overflow ได้แก่ คำสั่ง TRAPV

ฟังก์ชั่น Trace จะเริ่มทำเมื่อ T-bit ใน Status Register เป็น "1" โดยหลังจากทำคำสั่งหนึ่งใน User Mode แล้ว จะเกิด Exception ขึ้นทันที ช่วยให้สามารถทำ Single Step ได้

Exception จากภายนอก

Exception จากภายนอก เกิดได้จากการที่อุปกรณ์ภายนอกส่งสัญญาณต่อไปนี้เข้ามา

1) สัญญาณ $\overline{BERR}$

2) สัญญาณ $\overline{RESET}$

3) สัญญาณ Interrupt ($\overline{IP0}$, $\overline{IP1}$, $\overline{IP2}$)

ระดับความสำคัญของ Exception

เมื่อเกิด Exception ขึ้นหลายอย่างพร้อม ๆ กัน MC 68000 จะจัดการกับ Exception ที่มีระดับสูงสุดก่อน และในระหว่างการจัดการนั้นยอมให้มี Exception ที่สูงกว่ามา Exception ได้เท่านั้น ระดับความสำคัญของ Exception ได้แสดงไว้ในตารางด้านล่าง

จาก**รูปที่ 9** Exception แบ่งเป็นชุดใหญ่ ๆ ได้ 3 ชุด ชุดที่มีระดับความสำคัญสูงที่สุด, รองลงมา และต่ำสุด ชุดที่มีความสำคัญสูงที่สุดจะ **หยุดการทำคำสั่งทันทีเมื่อเกิด Exception ในชุดนี้**

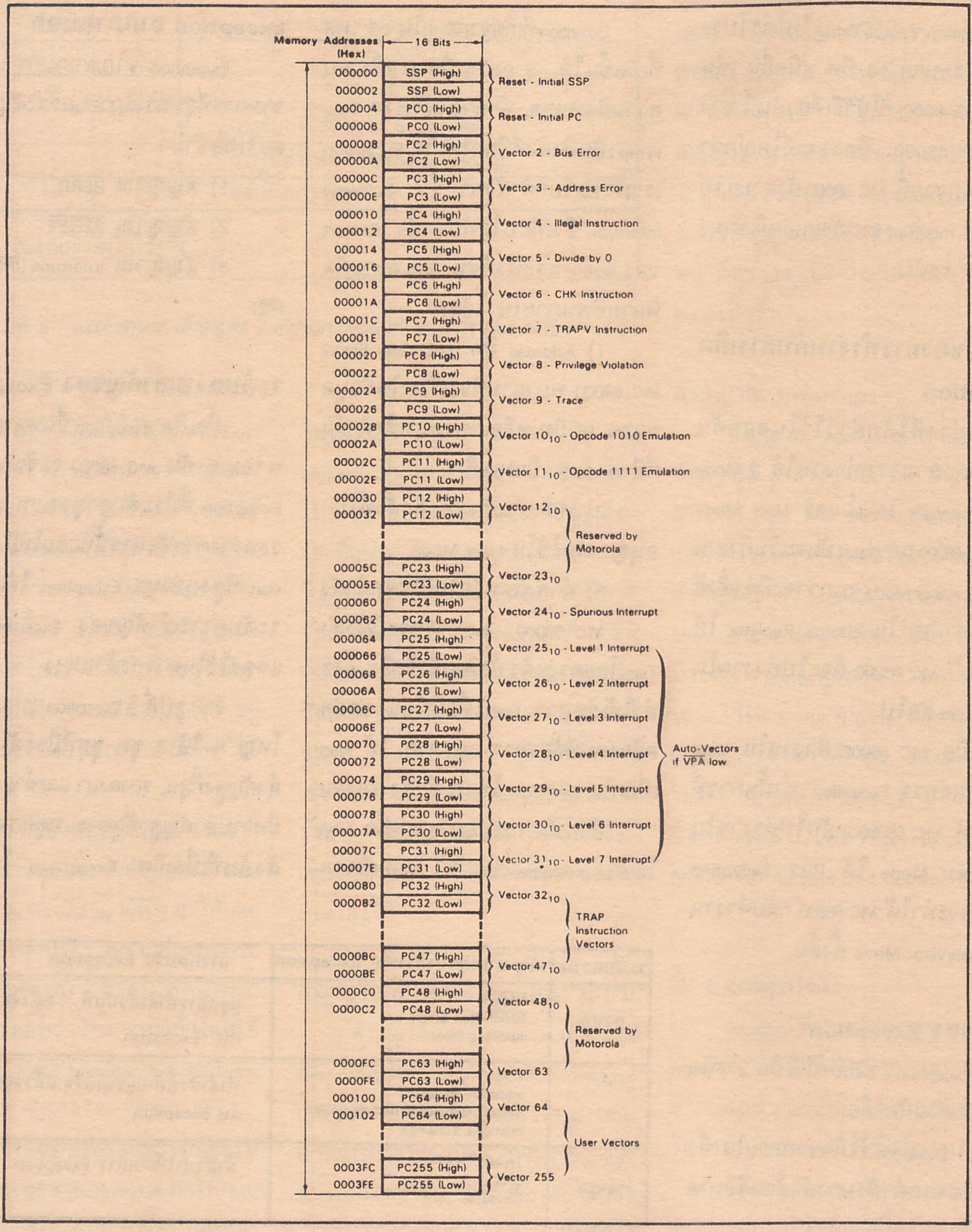
ระดับความสำคัญ	สาเหตุของการเกิด Exception	การตอบรับ Exception
สูงสุด	$\overline{RESET}$ $\overline{BERR}$ (Bus Error) Address Error	หยุดการทำคำสั่งทันที แล้วจัดการกับ Exception
	Trace Interrupt Request Illegal/Unimplemented Opcode Privilege Violation	ทำคำสั่งที่ทำอยู่จนเสร็จ แล้วจัดการกับ Exception
ต่ำสุด	TRAP, TRAPV CHK Divide-by-zero	คำสั่งทำให้เกิดการ Exception

รูปที่ ๙ Exception ชนิดต่าง ๆ และการตอบรับ

ขึ้น โดยไม่รอให้ทำคำสั่งนั้นเสร็จ ชุดถัดมาจะรอให้ทำจนจบคำสั่งก่อน จึงเริ่มทำการ Exception ซึ่ง Exception ในชุดนี้รวมถึงการ Interrupt ด้วย ซึ่งในการ Interrupt ยังมีการแบ่งระดับความสำคัญอีก 7 ระดับ Exception ชุดสุดท้ายก็จะรอให้จบคำสั่งก่อนเช่นเดียวกัน

Exception Vector Table

ศูนย์กลางของการจัดการ Ex-



รูปที่ 10 Exception Vector Table

ception อยู่ที่ตารางนี้ ตารางนี้มีขนาด 1024 ไบท์ อยู่ใน Memory Address ที่ $(000000)_{16}$ ถึง $(0003FF)_{16}$ ดังที่แสดงไว้ในรูปที่ 10

ตารางนี้ประกอบด้วย Address ขนาด 32 บิท (เพื่อที่จะ Load เข้า Program Counter ได้โดยตรง) จำนวน 256 Address จากตารางจะเห็นได้ว่า Exception แต่ละชนิดจะมี Address ที่แน่นอนที่ใช้เก็บ Address ของ Routine ที่ใช้จัดการ Exception นั้น ๆ ตารางยังแบ่งเป็น 2 ส่วน คือ ส่วนที่กำหนดไว้แล้ว สำหรับ Exception ชนิดต่าง ๆ ซึ่งได้แก่ 64 Address (Vector) แรก และส่วนที่เหลืออีก 192 Address (Vector) นั้น User สามารถกำหนดได้เอง นอกจากนี้แล้ว จะเห็นได้ว่ามีบาง Address ที่ถูกสำรองไว้โดยบริษัท Motorola ผู้ผลิตเอง

อุปกรณ์ภายนอก สามารถใช้ Adress ใน User Vector ได้โดยเมื่อส่งสัญญาณ Interrupt เข้ามาแล้ว MC 68000 จะตอบโดยส่ง Interrupt Acknowledge (ใช้สัญญาณ Function Code และมีสัญญาณ A_1 - A_3 แสดงระดับความสำคัญที่ Interrupt เข้ามา) อุปกรณ์ก็จะส่งสัญญาณ $\overline{DTACK}$ พร้อมกับเลขที่ Vector มาตามสาย D0-D7 แต่ถ้าไม่ต้องการกำหนดเลขที่ Vector เองก็ส่งสัญญาณ $\overline{VPA}$ มา MC 68000 ก็จะทำ Routine ตามที่กำหนดไว้แล้ว สำหรับ Interrupt ระดับต่าง ๆ การที่ User สามารถกำหนด Vector ได้เองนี้ ทำให้ MC 68000 สามารถทำ Exception ได้มากกว่า 200 ชนิด

Table 2-1. Addressing Mode Summary

Mode	Address Formation
Register Direct Addressing Data Register Direct Address Register Direct	 EA = DREGn EA = AREGn
Register Indirect Addressing Register Indirect Postincrement Register Indirect Predecrement Register Indirect Register Indirect with Offset Indexed Register Indirect with Offset	 EA = (AREGn) EA = (AREGn); Increment (AREGn) Decrement (AREGn); EA = (AREGn) EA = (AREGn) + data16 EA = (AREGn) + (XREGn) + data8
Implied Register Addressing	EA = SR, SP, PC
Absolute Addressing Absolute Short Absolute Long	 EA = (Next word) EA = (Next 2 words)
Program Counter Relative Addressing Relative with Offset Relative with Index and Offset	 EA = (PC) + data16 EA = (PC) + (XREG) + data8
Immediate Data Addressing Immediate Quick Immediate	 Data = Next word or words Data inherent in instruction word
EA = Effective Address DREGn = Any Data Register AREGn = Any Address Register XREGn = Any Data or Address Register used as an Index Register ( ) = Contents of	data8 = 8-bit offset (displacement) data16 = 16-bit offset (displacement) SR = Status Register SP = Stack Pointer (User or Supervisor) PC = Program Counter

รูปที่ 11 สรุป Addressing Mode ต่าง ๆ ของ 68000

4. Addressing Mode

MC 68000 มีวิธี Address ได้ทั้งหมด 6 ชนิด ซึ่งได้แสดงไว้ในรูปที่ 11

1) Register Direct Addressing เป็นการกำหนด Register ที่จะใช้งานโดยตรง

2) Register Indirect Addressing วิธีการ Address ชนิดนี้ ค่าของ Address ที่ต้องการจะอยู่ใน Address Register ซึ่งยังอาจจะมีการบวก หรือลบด้วยค่าคงที่ หรือค่าใน Index Register (Index Register จะเป็น Data Register หรือ Address Register ก็ได้)

3) Implied Register Addressing หมายถึงการใช้ Stack Pointer, Status Register หรือ Program Counter ซึ่งจำเป็นต้องใช้อยู่แล้วสำหรับคำสั่งนั้น ๆ

4) Absolute Addressing เป็นการกำหนด Address โดยตรงในคำสั่ง

5) Program Counter Relative Addressing เป็นการ Address โดยเทียบกับ Program Counter โดยบวกค่าใน Program Counter ด้วยค่าใน Index Register หรือค่าคงที่

6) Immediate Data Addressing เป็นการกำหนดข้อมูลโดยตรง อาจจะอยู่ใน OpCode เลยหรืออยู่ต่อจาก Opcode ก็ได้

5. ชุดคำสั่งของ 68000

MC 68000 มีคำสั่งพื้นฐานอยู่ 56 คำสั่งด้วยกัน ดังแสดงไว้ในรูปหน้าถัดไป) แต่เมื่อรวมกับการ Address อีกหลายชนิด ทำให้ชุดคำสั่งมีมากกว่า 300 คำสั่ง

เนื่องจาก MC 68000 ไม่มีคำสั่งเฉพาะสำหรับอุปกรณ์อินพุทเอาท์พุท แต่ใช้วิธี Memory Mapped I/O ดังนั้นคำสั่งทุกคำสั่งที่อ้างถึง Memory ก็ใช้เป็นคำสั่งอินพุท เอาท์พุท ได้เพราะ MC 68000 จะถือว่า อุปกรณ์อินพุท เอาท์พุท เสมือน Memory ตำแหน่งหนึ่ง

Table 6-3. The 68000 Instruction Set

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
Bcc	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Compare
DBcc	Test Cond., Decrement and Branch
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral Data
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PEA	Push Effective Address
RESET	Reset External Devices
ROL	Rotate Left without Extend
ROR	Rotate Right without Extend
ROXL	Rotate Left with Extend
ROXR	Rotate Right with Extend
RTE	Return from Exception
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditional
STOP	Stop
SUB	Subtract
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink

รูปที่ 12 ตารางสรุปคำสั่งพื้นฐาน ของ 6800

รูปแบบของคำสั่งของ MC 68000 จะเริ่มด้วย Opcode ขนาด 2 ไบท์ เสมอ ส่วนคำสั่งทั้งหมดอาจจะยาวได้ตั้งแต่ 2 ไบท์ ถึง 10 ไบท์

คำสั่งส่วนใหญ่ที่อ้างถึง Memory สามารถที่จะอ้างขนาด 1 Byte, Wopd (2 Byte) หรือ long Word (4 ไบท์) โดยในคำสั่งจะมี .B.W และ .L ต่อท้ายตามลำดับ

MC 68000 ไม่มีคำสั่งที่ใช้ย้ายข้อมูลเป็นบล็อค อย่างของ Z 8000 หรือ 8086 แต่ MC 68000 มีคำสั่งพื้นฐานที่ทำให้เขียนโปรแกรมที่ย้ายข้อมูลเป็นบล็อคได้ง่าย ๆ

MC 68000 มีคำสั่งสำหรับการคูณ และการหาร ซึ่ง Z 8000 ก็มี โดย Z 8000 สามารถคูณและหารได้เฉพาะแบบที่ไม่มีเครื่องหมายเท่านั้น ส่วน MC 68000 สามารถคูณและหารแบบมีหรือไม่มีเครื่องหมายก็ได้ แต่ Z 8000 สามารถคูณและหารขนาด 32 บิทได้ MC 68000 คูณและหารได้ แค่แบบ 16 บิทเท่านั้น

คำสั่งที่น่าสนใจของ MC 68000 ก็คือคำสั่ง LINK และ UNLK คำสั่งคู่นี้ใช้จองที่สำหรับจัดการเกี่ยวกับการส่งผ่านข้อมูลระหว่าง Subroutine ต่าง ๆ

สรุป

MC 68000 เป็นไมโครโปรเซสเซอร์ แบบ 16 บิท ชนิดแรก ซึ่งมีโครงสร้างภายในแบบ 32 บิท และสามารถที่จะ Address Memory ได้โดยตรงถึง 16 M หรือ 16,777,216 ตำแหน่ง Memory ขนาดนี้เท่ากับของเครื่อง mainframe IBM 370 ทีเดียว นอกจากนี้เมื่อพิจารณาดูถึงลักษณะสถาปัตยกรรมภายในแล้ว MC 68000 ใช้ register ขนาด 32 บิท ถึง 17 ตัว การใช้งาน register เหล่านี้ไม่มีการเจาะจงหน้าที่ให้โดยเฉพาะอย่างเช่นให้ register ตัวนั้นทั้งนี้เป็น Accumulator บ้างเป็น Index Register บ้าง ซึ่งทำให้สามารถเขียนโปรแกรมได้ง่ายขึ้น และมีประสิทธิภาพขึ้น

MC 68000 สามารถรับส่งข้อมูลได้ทั้งแบบ Asynchronous และ Synchronous ทำให้สามารถติดต่อได้กับทั้งอุปกรณ์ที่ทำงานเร็วและช้าโดยไม่ต้องมีวงจรเพิ่มเติมอีก และการที่บรรจุในชิพขนาด 64 ขา ทำให้ทั้งสัญญาณ Address และ Data ไม่ต้องใช้การ Multiplex จึงลดความซับซ้อนของวงจรภายนอกลงไปมากนอกจากนี้ยังสามารถที่จะทำการ Interrupt ได้ทั้งโดย Software และ Hardware ซึ่งเมื่อรวมกันแล้วจะมีมากกว่า 200 ชนิด ส่วนการอ้าง Address ก็ทำได้ถึง 14 แบบด้วยกัน

ชุดคำสั่งของ MC 68000 ประกอบด้วยคำสั่งพื้นฐาน 56 คำสั่ง แต่ละคำสั่งก็มีการขยายแยกย่อออกไปตามลักษณะของ Addressing mode ต่าง ๆ รวมแล้ว เป็นหลายร้อยคำสั่งด้วยกัน ซึ่งอำนวยความสะดวกในการใช้งานต่าง ๆ อย่างมาก

เมื่อพิจารณาถึงคุณลักษณะทั้งหมดของ MC 68000 แล้วก็นับว่าเป็นไมโครโปรเซสเซอร์ที่มีวิวัฒนาการไปไม่น้อย กำลังได้รับความนิยมเพิ่มขึ้นอย่างรวดเร็ว ในอนาคตอันใกล้นี้ MC 68000 อาจจะกลายเป็น CPU ที่ใช้กันแพร่หลายจนเป็นมาตรฐานในระดับ 16/32 บิทเช่นเดียวกันที่ 8088 เป็นมาตรฐานในระดับ 8/16 บิทไปก็ได้ ■

MC 68000

68000 VS 8086/8088

เพื่อให้บทความนี้สมบูรณ์ยิ่งขึ้นผมขอเปรียบเทียบลักษณะต่าง ๆ ของ MC 68000 กับ CPU ตระกูล 8086/8088 ของ Intel ซึ่งเป็น CPU ที่ใช้กันแพร่หลายที่สุดในปัจจุบันดังนี้ (รายละเอียดเพิ่มเติมของ 8086/8088 ดูได้ในไมโครฯ ฉบับที่ 8)

● Pipeline Architechture หมายถึงการทำงานตามขั้นตอนต่าง ๆ ใน Machine cycle ของ CPU โดยมีการต่อเนื่องหรือใช้ช่วงเวลาที่ทับซ้อนกัน 8086/8088 จะทำ pipeline นี้โดยแบ่งเป็น 2 ขั้นคือ ขั้นการ fetch เอาคำสั่งจาก memoryมา decode โดย BIU (Bus Interface Unit) กับขั้นการ execute โดย EU (Dxecution Unit) ซึ่ง 2 ขั้นนี้จะทำไปพร้อม ๆ กันได้ โดยขณะที่ EU ทำการ execute BIU จะไปทำการ fetch เอาคำสั่งต่อไปเข้ามาเก็บไว้ใน queue ภายใน CPU (ยาว 6 ไบท์สำหรับ 8086 และ 4 ไบท์ สำหรับ 8088) ส่วน 68000 จะ ...peline โดยแบ่งเป็น 3 ขั้นคือ ขั้นการ fetch, decode และ execute โดยขณะที่คำสั่งหนึ่งถูก execute อีกคำสั่งหนึ่งจะถูก decode คอยอยู่ และอีกคำสั่งหนึ่งกำลังถูก fetch เข้ามาจาก memory ดังนั้นการทำงานของ 68000 ระหว่างขั้นการ fetch กับ execute จึงล่วงหน้ากันอยู่ 2 คำสั่ง

● System Configuration โครงสร้างของระบบที่ใช้ 8086/8088 จะมี 2 ประเภทคือระบบเล็กกับระบบใหญ่โดยที่ตัว CPU จะมีสัญญาควบคุมให้ทำงานใน 2 mode คือ maximum mode สำหรับระบบใหญ่ที่มีส่วนประกอบของวงจรมาก ๆ หรือซับซ้อนแต่ละขาของ CPU ก็จะทำหน้าที่อย่างหนึ่ง ส่วนถ้าเป็น minimum mode ก็ใช้กับระบบเล็กที่มีส่วนประกอบไม่ซับซ้อน สัญญาณจากขาต่าง ๆ ของ CPU ก็จะเป็นอีกอย่างหนึ่งทั้งนี้เนื่องจาก 8086/8088 บรรจุในตัวถังไอซีขนาด 40 ขาเท่านั้น จึงต้องมีการ multiplex ให้ขาหนึ่งทำหน้าที่หลายอย่าง การที่ต้องแบ่งเป็น 2 mode ก็เพื่อลดการ multiplex สัญญาณที่ไม่จำเป็นสำหรับการทำงานใน mode หรือ configuration นั้น ๆ ออกไปส่วน 68000 จะทำงานในลักษณะของระบบใหญ่เสมอ เนื่องจาก 68000 บรรจุในตัวถัง 64 ขา จึงมีขามากพอสำหรับสัญญาณต่าง ๆ โดยไม่ต้องมีการ multiplex ดังนั้นสัญญาณต่าง ๆ จึงมีให้ใช้ได้ทั้งหมด ไม่แบ่งเป็น mode (mode ที่ว่านี้คือ mode ของ confignration อย่างสับสนกับการแบ่งการทำงานภายในเป็น User และ Supervisor mode) เพราะไม่มีความจำเป็นที่จะต้องลดขาและสัญญาณให้เหลือจำกัดเท่าที่จำเป็น ซึ่งการที่มีขามากนี้มีข้อเสียคือทำให้เปลืองเนื้อที่สำหรับ BUS ต่าง ๆ บนบอร์ด แต่ก็ลดวงจร Logic ภายนอกที่ต้องมีไว้ decode การ multiplex สัญญาณต่าง ๆ ที่ออกมาจากขาเดียวกัน ว่าตอนไหนเป็นสัญญาณอะไรอย่างในกรณีของ 8086/8088

● Address Space 68000 มีขีดความสามารถในการอ้าง Address (Address Space) ได้ถึง 16 M ผ่านทาง Address Bus 23 สาย และ $\overline{UDS}$ กับ $\overline{LDS}$ และยังอาจขยายออกได้ถึง 64 โดยอาศัยสัญญาณ Function code (FC1, FC2, FC3) ช่วย memory ของ 68000 ต่อเนื่องกันหมดไม่มีการแบ่งเป็น Segment ส่วน 8086/8088 นั้นมีความสามารถที่จะอ้างแอดเดรสได้ในขอบเขตของ Segment ขนาด 64K byte เท่านั้น (Program Counter มีขนาดเพียง 16 บิท, 2^{16} = 64K) แต่มี Segment register ช่วยให้อ้างถึง Segment ต่าง ๆ กันได้อีก 16 Segment รวมเป็น Address Space 64K × 16 = 1024K หรือ 1M)

● Operation Mode 68000 มีการทำงานภายในแบ่งเป็น 2 mode คือ Supervisor mode สำหรับ System program เช่น Operating System และ User mode สำหรับ application program ของ User ส่วน 8086/8088 ไม่มีลักษณะดังกล่าว (แต่การแบ่งการทำงานเป็น 2 mode เช่นนี้เหมือนกับใน Z-8000 ซึ่งเป็น CPU 16 บิทของบริษัท Zilog ผู้สร้าง Z-80 ซึ่งจะแบ่งการทำงานเป็น System mode และ Normal mode ซึ่งเทียบได้กับ Supervisor และ user mode ของ 68000 ตามลำดับ)

● Register 68000 มี register ขนาด 32 บิท ถึง 17 ตัว ในจำนวนนี้ 8 ตัวนี้แบ่งเป็น Data register ที่จะเรียกใช้ข้อมูลได้ทั้งทีละ 8, 16 และ 32 บิท ที่เหลืออีก 9 ตัวเป็น Address Register ที่ใช้ได้ทีละ 16 และ 32 บิท โดยมี 2 ตัวสงวนไว้สำหรับเป็น Stack Pointer ของ Supervisor และ User แต่ 68000 ไม่มี Register ที่ทำหน้าที่เฉพาะเป็น Accumulator ส่วนใน 8086/8088 มี Pegister ขนาด 16 บิทเพียง 12 ตัว แบ่งเป็น Accumulator 1 ตัว กับ general register อีก 3 ตัว กลุ่มหนึ่ง (เป็นกลุ่มเดียวที่สามารถเรียกได้ข้อมูลทีละ 8 บิทได้ด้วย) segment register 4 ตัวอีกกลุ่มหนึ่ง และ index กับ pointer อย่างละ 2 ตัว อีกกลุ่มหนึ่ง

เมื่อพิจารณาเปรียบเทียบกันแล้ว ระบบของ 68000 นั้นใหญ่และมีส่วนประกอบซับซ้อนกว่า 8086/8088 มาก จึงเป็นที่นิยมของไมโครคอมพิวเตอร์รุ่นใหม่ที่ต้องการให้มีขีดความสามารถสูงมี memory มาก ๆ มี System Software ที่ซับซ้อนเช่น window หรือintegrated Software เป็นต้น

“เรื่องยากที่อ่านไม่ยาก”

เปิดโลกคอมพิวเตอร์

วศิน เพิ่มทรัพย์



> *“สี่บิทเป็นหนึ่งนิบเบิ้ล*
> *สองนิบเบิ้ลเป็นหนึ่งไบท์*
> *สองไบท์เป็นหนึ่งเวิร์ด*
> *สองเวิร์ดเป็นหนึ่งดับเบิ้ลเวิร์ด”*
>
> **บทอาขยานยุคคอมพิวเตอร์**

ความจริงตั้งใจจะคุยกันเรื่องบิท ไบท์ เวิร์ด กันเลยอย่างที่นำไว้นั่นแหละครับ แต่มาคิดดูอีกทีก่อนจะออกมาเป็น บิท ไบท์ เวิร์ด อันเป็นโครงสร้างพื้นฐานของข้อมูลในระบบคอมพิวเตอร์นั้น ยังมีเรื่องของตรรกศาสตร์ หรือลอจิก (Logic) เข้ามาเกี่ยวข้องด้วยไม่น้อยเลย ดังนั้นเรามาแวะสนใจกับเรื่องของตรรกศาสตร์กับคอมพิวเตอร์เสียหน่อยก่อนดีกว่า

ในอดีตนั้น เครื่องจักรคำนวณใด ๆ ที่เกิดขึ้นจากความอุตสาหะของอัจฉริยะทั้งหลายนั้น ล้วนแต่เป็นกลไก หรือ Mechanical ทั้งสิ้น ซึ่งเวลาที่แทนค่าตัวเลขลงไปในกระบวนการคำนวณของเครื่องเหล่านั้นก็มักจะแทนด้วยเลขฐานสิบ หรือสัญลักษณ์อันมีพื้นฐานมาจากเลขฐานสิบที่มนุษย์เราใช้อยู่ทั้งสิ้น ไม่ว่าจะเป็นลูกคิดของจีนโบราณหรือเฟืองทดในเครื่องคำนวณสมัยหลัง ทั้ง ๆ ที่ความจริงพีชคณิตของตรรกศาสตร์หรือพีชคณิตของบูล (Boolean algebra) อันเป็นรากฐานของระบบการทำงานในดิจิตอลคอมพิวเตอร์ปัจจุบันนั้นก็ได้ค้นพบกันมานานแล้ว โดย-George Boole นักคณิตศาสตร์ชาวอังกฤษ แต่ก็ไม่มีใครสนใจ นำมาใช้ในเครื่องคำนวณอย่างจริงจัง

จวบจนกระทั่งโลกศตวรรษที่ 20 ได้พัฒนาเข้าสู่ยุค

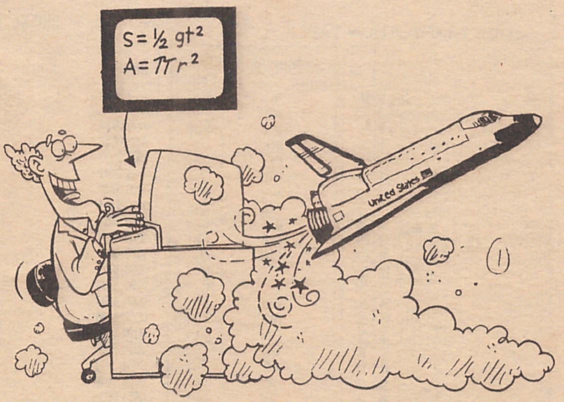
ไฟฟ้า วงจรไฟฟ้าที่มีเพียง 2 สถานะคือ ปิดกับเปิดได้ถูกนำมาใช้ในคอมพิวเตอร์ พีชคณิตของบูล จึงได้ถูกนำกลับมาใช้อีกครั้ง เริ่มต้นจากการค้นคว้าของ Claude E. Shannon แห่ง MIT ในเรื่อง A Symbolic Analysis of Relay and Switching Circuit ใน ค.ศ.1937 ซึ่งเป็นการนำแนวความคิดทางตรรกศาสตร์มาประยุกต์ใช้กับวงจรไฟฟ้าเป็นครั้งแรก

บิท-ไบท์

คุณที่เคยเรียนตรรกศาสตร์มาแล้วคงจะพอนึกออกถึงสภาวะทางตรรกศาสตร์ ซึ่งมีจำนวนอยู่เพียง 2 ค่า คือ เท็จกับจริงเท่านั้น จำนวนนี้แทนได้พอดีกับเลขฐานสองที่มีเพียง 2 ค่าเหมือนกันคือ 0 กับ 1 เท่านั้น (ดังนั้นในทางปฏิบัติเราก็เลยมักจะแทนค่าเท็จกับจริงด้วย 0 กับ 1 เพราะเขียนง่ายกว่ากันตั้งเยอะ) และพลอยเข้ากันได้พอดีกับสภาวะของวงจรไฟฟ้าที่มีเพียง 2 คือ ปิดวงจร (มีกระแสไหล) กับเปิดวงจร (ไม่มีกระแสไหล) เหมือนกัน ดังนั้นระบบทั้งสามนี้จึงสมนัยกัน และมักจะใช้แทนกันได้เสมอในระบบคอมพิวเตอร์ และนี่เองคือหน่วยข้อมูลที่เล็กที่สุดในระบบคอมพิวเตอร์ที่เราเรียกกันว่า บิท (bit = binary digit หมายถึงค่าที่แทนได้ด้วยเลขฐานสองหนึ่งหลัก)

แต่ในการใช้งานจริงนั้น ข้อมูลเพียงบิทเดียวก็ไม่อาจทำอะไรได้มากพอ ตัวอย่างเช่น คุณต้องการเก็บข้อมูลในระบบคอมพิวเตอร์เพื่อบันทึกว่า การโยนเหรียญของคุณ 1 เหรียญจะออกหัวหรือก้อย คุณก็อาจให้ 1 แทนการที่เหรียญออกหัว 0 แทน การที่เหรียญออกก้อยได้ แต่ถ้าคุณโยนเหรียญพร้อมกัน 2 เหรียญล่ะ คุณใช้บิทเดียวไม่พอเสียแล้ว ก็ต้องเพิ่มเป็น 2 บิท โดยให้บิทหนึ่งแทนเหรียญแรก อีกบิทหนึ่งแทนเหรียญที่สอง ก็จะแทนเหตุการณ์หรือบันทึกข้อมูลของเหตุการณ์ที่เกิดขึ้นได้ 4 อย่าง คือเหรียญออกหัว/หัว (1/1), ก้อย/ก้อย (0/0), หัว/ก้อย (0/1), ก้อย/หัว (1/0) เห็นไหมครับว่า เพียงเรื่องเล็กแค่นี้ก็ต้องใช้มากกว่า 1 บิทเสียแล้ว ดังนั้นจึงมีความจำเป็นจะต้องรวมหลาย ๆ บิทเข้าด้วยกันให้เป็น กลุ่มของข้อมูลที่ใหญ่กว่า เช่น 4 บิท เป็น 1 นิบเบิ้ล (nibble) ก็จะแทนข้อมูลได้ $2^4 = 16$ อย่าง (ไม่เชื่อคุณก็ลองเขียนเลขฐานสอง 4 บิทดู จะเขียนได้ต่างกัน = 16 แบบเท่านั้นเองครับ)

หน่วยของข้อมูลที่เรียกว่า nibble นี้ก็ยังไม่ค่อยจะเพียงพอกับความต้องการนัก ถึงแม้ว่าจะมีความแตกต่างกันมากพอจะแทนตัวเลขทั้ง 10 ตัว ที่มนุษย์เราใช้กันอยู่ได้แล้วก็ตาม เพราะงานคอมพิวเตอร์นั้นไม่ใช่จำกัดอยู่แต่ใน



คอมพิวเตอร์ในยุคแรกมักจะถูกใช้ในงานวิทยาศาสตร์

แวดวงของวิทยาศาสตร์ แต่ได้มีการขยายตัวไปสู่วงการธุรกิจและอื่น ๆ อีกมากมาย ดังนั้นจึงมีความจำเป็นที่จะแทนตัวอักษรทั้งตัวใหญ่และตัวเล็ก (ภาษาอังกฤษ) ให้เป็นข้อมูลของเครื่องได้ด้วย ดังนั้นจึงมีการรวมเอา 8 บิท หรือ 2 นิบเบิ้ลเข้าด้วยกันเป็น 1 ไบท์ (byte) ซึ่งก็จะแทนอักษร ตัวเลข ตลอดจนเครื่องหมายต่าง ๆ ได้ถึง $2^8 = 256$ แบบ ซึ่งมากพอใช้ เพราะอักษรภาษาอังกฤษมี 26 ตัว รวมตัวใหญ่-เล็ก และสัญลักษณ์พิเศษต่าง ๆ เช่น +-=?,. และอื่น ๆ แล้วก็ร้อยกว่าตัวเท่านั้น ยังมีที่เหลือไว้เผื่อเพิ่มเติมอะไรได้อีกมากทีเดียว

EBCDIC VS ASCII

เมื่อมีการแทนข้อมูลตัวอักษรแต่ละตัวด้วย 1 ไบท์แล้ว ก็ยังอุตส่าห์มีปัญหาเกิดขึ้นอีกจนได้ เพราะการกำหนดว่ารหัสหรือค่าใดตรงกับตัวอักษรใดนั้น มีผลต่อการออกแบบวงจร การทำงานภายในเครื่องคอมพิวเตอร์ และการเขียนโปรแกรมไม่น้อยเลย ดังนั้นถ้าต่างคนต่างออกแบบวงจร ต่างคนต่างพัฒนาโปรแกรม ต่างคนต่างก็จะตั้งรหัสของตัวเอง (แบบเดียวกับปัญหาเรื่องรหัสภาษาไทยที่ยังตกลงกันไม่ได้ในเวลานี้นั่นแหละครับ) ซึ่งมักจะอ้างว่าวิธีการหรือรหัสของตัวเองดีที่สุด ซึ่งปัญหาที่จะตามมาก็คือความเข้ากันไม่ได้ (incompatibility) ระหว่างระบบของผู้ผลิตแต่ละราย ทำให้การใช้งานมีปัญหายุ่งยากมากเรื่อง เช่น ข้อมูลจากคอมพิวเตอร์เครื่องหนึ่งจะไปใช้กับเครื่องที่มีรหัสไม่เหมือนกันไม่ได้ ดังนั้นในสหรัฐฯ จึงมีการกำหนดมาตรฐานขึ้นอันหนึ่ง เรียกว่า ASCII ย่อมาจาก American Standard Code for Information Interchange เป็นรหัสที่

CODE TRANSLATION TABLE

Dec.	Hex	EBCDIC(1)	ASCII	Binary
0	00	NUL	NUL	0000 0000
1	01	SOH	SOH	0000 0001
2	02	STX	STX	0000 0010
3	03	ETX	ETX	0000 0011
4	04	PF	EOT	0000 0100
5	05	HT	ENQ	0000 0101
6	06	LC	ACK	0000 0110
7	07	DEL	BEL	0000 0111
8	08	GE	BS	0000 1000
9	09	RLF	HT	0000 1001
10	0A	SMM	LF	0000 1010
11	0B	VT	VT	0000 1011
12	0C	FF	FF	0000 1100
13	0D	CR	CR	0000 1101
14	0E	SO	SO	0000 1110
15	0F	SI	SI	0000 1111
16	10	DLE	DLE	0001 0000
17	11	DC1	DC1	0001 0001
18	12	DC2	DC2	0001 0010
19	13	TM	DC3	0001 0011
20	14	RES	DC4	0001 0100
21	15	NL	NAK	0001 0101
22	16	BS	SYN	0001 0110
23	17	IL	ETB	0001 0111
24	18	CAN	CAN	0001 1000
25	19	EM	EM	0001 1001
26	1A	CC	SUB	0001 1010
27	1B	CU1	ESC	0001 1011
28	1C	IFS	FS	0001 1100
29	1D	IGS	GS	0001 1101
30	1E	IRS	RS	0001 1110
31	1F	IUS	US	0001 1111
32	20	DS	SP	0010 0000
33	21	SOS	¦ !	0010 0001
34	22	FS	"	0010 0010
35	23		#	0010 0011
36	24	BYP	$	0010 0100
37	25	LF	%	0010 0101
38	26	ETB	&	0010 0110
39	27	ESC	'	0010 0111
40	28		(	0010 1000
41	29		)	0010 1001
42	2A	SM	*	0010 1010
43	2B	CU2	+	0010 1011
44	2C		,	0010 1100
45	2D	ENQ	-	0010 1101
46	2E	ACK	.	0010 1110
47	2F	BEL	/	0010 1111
48	30		0	0011 0000
49	31		1	0011 0001
50	32	SYN	2	0011 0010
51	33		3	0011 0011
52	34	PN	4	0011 0100
53	35	RS	5	0011 0101
54	36	UC	6	0011 0110
55	37	EOT	7	0011 0111
56	38		8	0011 1000
57	39		9	0011 1001
58	3A		:	0011 1010
59	3B	CU3	;	0011 1011
60	3C	DC4	<	0011 1100
61	3D	NAK	=	0011 1101
62	3E		>	0011 1110
63	3F	SUB	?	0011 1111

CODE TRANSLATION TABLE

Dec.	Hex	EBCDIC(1)	ASCII	Binary
64	40	Sp	@	0100 0000
65	41		A	0100 0001
66	42		B	0100 0010
67	43		C	0100 0011
68	44		D	0100 0100
69	45		E	0100 0101
70	46		F	0100 0110
71	47		G	0100 0111
72	48		H	0100 1000
73	49		I	0100 1001
74	4A	¢	J	0100 1010
75	4B	.	K	0100 1011
76	4C	<	L	0100 1100
77	4D	(	M	0100 1101
78	4E	+	N	0100 1110
79	4F	|	O	0100 1111
80	50	&	P	0101 0000
81	51		Q	0101 0001
82	52		R	0101 0010
83	53		S	0101 0011
84	54		T	0101 0100
85	55		U	0101 0101
86	56		V	0101 0110
87	57		W	0101 0111
88	58		X	0101 1000
89	59		Y	0101 1001
90	5A	!	Z	0101 1010
91	5B	$	[	0101 1011
92	5C	*	\	0101 1100
93	5D	)	]	0101 1101
94	5E	;	¬ ^	0101 1110
95	5F	¬	_	0101 1111
96	60	-	`	0110 0000
97	61	/	a	0110 0001
98	62		b	0110 0010
99	63		c	0110 0011
100	64		d	0110 0100
101	65		e	0110 0101
102	66		f	0110 0110
103	67		g	0110 0111
104	68		h	0110 1000
105	69		i	0110 1001
106	6A	¦	j	0110 1010
107	6B	,	k	0110 1011
108	6C	%	l	0110 1100
109	6D	_	m	0110 1101
110	6E	>	n	0110 1110
111	6F	?	o	0110 1111
112	70		p	0111 0000
113	71		q	0111 0001
114	72		r	0111 0010
115	73		s	0111 0011
116	74		t	0111 0100
117	75		u	0111 0101
118	76		v	0111 0110
119	77		w	0111 0111
120	78		x	0111 1000
121	79		y	0111 1001
122	7A	:	z	0111 1010
123	7B	#	{	0111 1011
124	7C	@	|	0111 1100
125	7D	'	}	0111 1101
126	7E	=	~	0111 1110
127	7F	"	DEL	0111 1111

ตารางที่ 1 รหัส ASCII และ EBCDIC

ผู้ผลิตคอมพิวเตอร์ทั่วไปยอมรับไปใช้กัน รหัส ASCII นี้กำหนดตามความจำเป็นคือ มีตัวอักษรและสัญลักษณ์ต่าง ๆ ร้อยตัวเศษ ก็กำหนดไว้เพียง 7 บิท ซึ่งจะแทนอักษรได้ $2^7 = 128$ แบบ ก็พอแล้ว ส่วนเรื่องที่ใครจะกำหนดนอกเหนือจากนั้นไปก็เป็นเรื่องของผู้ผลิตแต่ละราย ไม่เป็นมาตรฐานอันเดียวกัน รหัส ASCII นี้ก็เป็นดังในตารางที่ 1

จากตาราง รหัสที่เป็นมาตรฐานจริง ๆ นั้นจะเริ่มตั้งแต่ 32 ฐานสิบ (ถ้าเขียนเป็นเลขฐานสองก็คือ 010 0000 หรือแปลงเป็นฐานสิบหกจะได้ 20) จนถึง 127 ฐานสิบ (111 1111 หรือ 7F ฐานสิบหก) ส่วนที่ต่ำกว่า 32 ลงไปก็เป็นรหัสที่ใช้ในการควบคุมทางฮาร์ดแวร์ของเครื่อง ซึ่งขึ้นอยู่กับการออกแบบของบริษัทแต่ละราย แต่ก็มักจะใกล้เคียงกันพอสมควร เช่น คีย์ ESC หรือ Escape ของแต่ละเครื่องจะแทนด้วยรหัส 27, RETURN หรือ ENTER แทนด้วย 13 เป็นต้น ส่วนที่เกิน 127 ไปนั้นก็เป็นเรื่องที่จะต้องดัดแปลงกันไปอีกทีหนึ่ง เช่นการแทนรหัสภาษาไทยลงในคอมพิวเตอร์ก็มักจะใช้รหัสส่วนที่เกิน 127 นี้ โดยเพิ่มบิทที่ 8 เข้าไป เพราะรหัส 7 บิทนั้นมีค่าสูงสุดแค่ 127 เท่านั้น

ที่จริงก็น่าจะหมดปัญหาแค่นั้น แต่ปรากฏว่ามีผู้ผลิตคอมพิวเตอร์รายใหญ่รายหนึ่งไม่ยอมรับรหัส ASCII นี้

CODE TRANSLATION TABLE

Dec.	Hex	EBCDIC(1)	ASCII	Binary
128	80			1000 0000
129	81	a		1000 0001
130	82	b		1000 0010
131	83	c		1000 0011
132	84	d		1000 0100
133	85	e		1000 0101
134	86	f		1000 0110
135	87	g		1000 0111
136	88	h		1000 1000
137	89	i		1000 1001
138	8A			1000 1010
139	8B			1000 1011
140	8C			1000 1100
141	8D			1000 1101
142	8E			1000 1110
143	8F			1000 1111
144	90			1001 0000
145	91	j		1001 0001
146	92	k		1001 0010
147	93	l		1001 0011
148	94	m		1001 0100
149	95	n		1001 0101
150	96	o		1001 0110
151	97	p		1001 0111
152	98	q		1001 1000
153	99	r		1001 1001
154	9A			1001 1010
155	9B			1001 1011
156	9C			1001 1100
157	9D			1001 1101
158	9E			1001 1110
159	9F			1001 1111
160	A0			1010 0000
161	A1	~		1010 0001
162	A2	s		1010 0010
163	A3	t		1010 0011
164	A4	u		1010 0100
165	A5	v		1010 0101
166	A6	w		1010 0110
167	A7	x		1010 0111
168	A8	y		1010 1000
169	A9	z		1010 1001
170	AA			1010 1010
171	AB			1010 1011
172	AC			1010 1100
173	AD			1010 1101
174	AE			1010 1110
175	AF			1010 1111
176	B0			1011 0000
177	B1			1011 0001
178	B2			1011 0010
179	B3			1011 0011
180	B4			1011 0100
181	B5			1011 0101
182	B6			1011 0110
183	B7			1011 0111
184	B8			1011 1000
185	B9			1011 1001
186	BA			1011 1010
187	BB			1011 1011
188	BC			1011 1100
189	BD			1011 1101
190	BE			1011 1110
191	BF			1011 1111

CODE TRANSLATION TABLE

Dec.	Hex	EBCDIC(1)	ASCII	Binary
192	C0	{		1100 0000
193	C1	A		1100 0001
194	C2	B		1100 0010
195	C3	C		1100 0011
196	C4	D		1100 0100
197	C5	E		1100 0101
198	C6	F		1100 0110
199	C7	G		1100 0111
200	C8	H		1100 1000
201	C9	I		1100 1001
202	CA			1100 1010
203	CB			1100 1011
204	CC	ʃ		1100 1100
205	CD			1100 1101
206	CE	Y		1100 1110
207	CF			1100 1111
208	D0	}		1101 0000
209	D1	J		1101 0001
210	D2	K		1101 0010
211	D3	L		1101 0011
212	D4	M		1101 0100
213	D5	N		1101 0101
214	D6	O		1101 0110
215	D7	P		1101 0111
216	D8	Q		1101 1000
217	D9	R		1101 1001
218	DA			1101 1010
219	DB			1101 1011
220	DC			1101 1100
221	DD			1101 1101
222	DE			1101 1110
223	DF			1101 1111
224	E0	\		1110 0000
225	E1			1110 0001
226	E2	S		1110 0010
227	E3	T		1110 0011
228	E4	U		1110 0100
229	E5	V		1110 0101
230	E6	W		1110 0110
231	E7	X		1110 0111
232	E8	Y		1110 1000
233	E9	Z		1110 1001
234	EA			1110 1010
235	EB			1110 1011
236	EC	⑁		1110 1100
237	ED			1110 1101
238	EE			1110 1110
239	EF			1110 1111
240	F0	0		1111 0000
241	F1	1		1111 0001
242	F2	2		1111 0010
243	F3	3		1111 0011
244	F4	4		1111 0100
245	F5	5		1111 0101
246	F6	6		1111 0110
247	F7	7		1111 0111
248	F8	8		1111 1000
249	F9	9		1111 1001
250	FA	|		1111 1010
251	FB			1111 1011
252	FC			1111 1100
253	FD			1111 1101
254	FE			1111 1110
255	FF	EO		1111 1111

นั่นคือ IBM ซึ่งผลิตเครื่องระดับเมนเฟรมจำนวนมาก และทุกเครื่องใช้รหัสแบบของ IBM ที่เรียกว่า EBCDIC (ย่อมาจาก Extended Binary Coded Decimal Interchanged Code) เนื่องจาก IBM เป็นบริษัทใหญ่และมีผลิตภัณฑ์ ของตนเองครบทั้งระบบ ดังนั้นจึงใช้อุปกรณ์ของตนเองโดยตลอด ไม่ต้องอาศัยอุปกรณ์ประกอบที่ใช้รหัสแบบอื่น ทำให้ระบบของ IBM ทำงานได้โดยไม่เกิดปัญหาเรื่อง incompatibility ที่กล่าวแล้ว จวบจนกระทั่งเมื่อปี 1982 ซึ่ง IBM กระโดดเข้าสู่ไมโครคอมพิวเตอร์ด้วยการออกเครื่อง IBM PC ที่เรารู้จักกันดี จึงได้ยอมนำรหัส ASCII ไปใช้กับเครื่องดังกล่าว เพื่อให้เข้ากับอุปกรณ์อื่น ๆ ในตลาดไมโครคอมพิวเตอร์ได้ รหัส EBCDIC นี้มีกำหนดไว้ทั้ง 8 บิท โดยส่วนที่เป็นตัวอักษรจะมีค่าอยู่เกินกว่า 127 ขึ้นไป ดังในตารางที่ 1

เวิร์ด/ฮาล์ฟเวิร์ด/ดับเบิลเวิร์ด

ทีนี้ก็มาถึงอีกคำหนึ่งคือเวิร์ด ซึ่งเป็นหน่วยของข้อมูลที่ใหญ่ถัดจากไบท์ขึ้นมา และเป็นขนาดที่ก่อให้เกิดความสับสนได้มากที่สุด เพราะไม่มีมาตรฐานที่กำหนดแน่นอนได้ว่า หนึ่งเวิร์ดจะมีขนาดเท่าไหร่ เหตุที่ต้องมีหน่วยของข้อมูลขนาดเวิร์ดก็เนื่องมาจากการคำนวณทางคณิตศาสตร์นั่นเอง

ในการแทนตัวเลขลงในระบบคอมพิวเตอร์นั้น ดังที่กล่าวแล้วในตอนแรกว่าเราอาจแทนได้ 1 ตัวต่อ 1 ไบท์ เช่น 173 แทนด้วยรหัส ASCII ใช้เนื้อที่ 3 ไบท์ แทน 1/7/3

เขียนเป็นเลขฐานสิบดังนี้คือ 49/55/51 (/ คั่นระหว่างข้อมูลแต่ละไบท์) ถ้าแทนด้วยวิธีนี้จะนำไปคำนวณได้ยาก เพราะต้องคิดเป็นเลขฐานสิบแบบที่มนุษย์คิด ในไมโครทั่ว ๆ ไปจะทำไม่ได้ (นอกจากผู้ใช้จะเขียนโปรแกรมขึ้นเองแต่จะทำงานช้า) กับอีกวิธีหนึ่ง ใช้แทนเป็นเลขฐานสองที่มีค่าตามเลขนั้น ๆ เลย คือ แทนด้วยเลขฐานสอง 1010 1101 เวลานำไปคำนวณก็ทำได้พร้อม ๆ กันทั้ง 8 บิทเลย เพราะว่าวงจรภายในเครื่องคอมพิวเตอร์สามารถทำได้โดยไม่ยุ่งยากนัก เท่ากับเป็นการคำนวณเลขทั้งสามหลักพร้อม ๆ กันไป แต่ถ้าใช้แค่ 8 บิทก็จะแทนได้แค่เลขตั้งแต่ 0 ถึง 2^8-1 = 255 (0000 0000 ถึง 1111 1111) เท่านั้น จึงต้องใช้มากกว่า 1 ไบท์ขึ้นไปเพื่อให้ได้ตัวเลขมากพอ กล่าวคือ สองไบท์จะแทนได้ตั้งแต่ 0 ถึง 2^{16}-1 คือ 65, 535 หรือถ้าใช้ 4 ไบท์ก็แทนได้ตั้งแต่ 0 ถึง 2^{32} = 4,294,967,295 ซึ่งมากพอสำหรับการคำนวณทั่ว ๆ ไป ดังนั้นจึงมีการจัดหน่วยข้อมูลเป็นเวิร์ด ขึ้นประกอบด้วยข้อมูลหลาย ๆ ไบท์รวมกัน เช่น ในไมโครโปรเซสเซอร์ 68000 จะใช้ 2 ไบท์ = 1 เวิร์ด และ 4 ไบท์ = 1 ลองเวิร์ด (longword) ส่วนในเครื่องเมนเฟรม IBM 370 นั้น 2 ไบท์จะเป็น 1 ฮาล์ฟเวิร์ด, 4 ไบท์เป็น 1 เวิร์ด และ 8 ไบท์เป็น 1 ดับเบิ้ลเวิร์ด แต่สำหรับไมโครโปรเซสเซอร์ที่มีโครงสร้างเป็นแบบ 8 บิทนั้น ก็จะไม่มีการแบ่งหน่วยของข้อมูลเป็น เวิร์ด แต่ก็มีคำสั่งให้ทำการคำนวณกับข้อมูลทีละ 2 ไบท์ได้ เช่น คำสั่ง DAD (Doublebyte ADd) ของ 8080 เป็นต้น

Sign bit

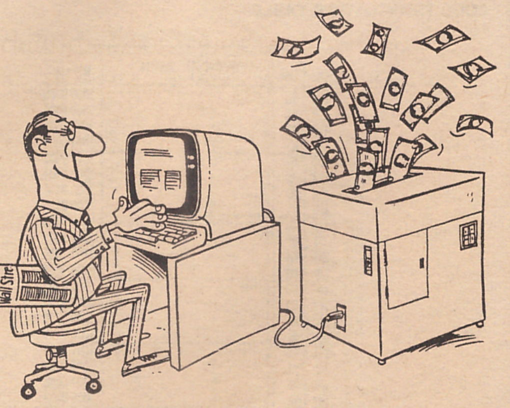
การแทนข้อมูลตัวเลขเพื่อใช้ในการคำนวณ ยังมีปัญหาอีกอย่างหนึ่งเข้ามาเกี่ยวข้อง คือสัญลักษณ์ +- หน้าตัวเลขเหล่านั้น เช่น -1,350 กับ +1,350 จะต้องแทนด้วยวิธีการที่แตกต่างกัน และเนื่องจากเครื่องหมายดังกล่าวก็มีเพียง 2 แบบ คือ + กับ - เท่านั้น เราจึงสามารถใช้ 1 บิทแทนเครื่องหมายได้ ที่เราเรียกว่า Sign bit นั่นเอง โดยนิยมให้เลขบวกมี Sign bit เป็น 0 เลขลบมี Sign bit เป็น 1 ดังนั้นเลข +135 จึงเก็บในเนื้อที่ 2 ไบท์ (16 บิท) เป็น

sign

0	0	0	0	0	1	0	1	0	1	0	0	0	1	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ไบท์หน้า ไบท์หลัง

ส่วนเลขลบนั้น sign bit ก็เป็น 1 แต่วิธีการเก็บค่าของมันจะยุ่งยากขึ้นเล็กน้อย กล่าวคือ เก็บเป็นแบบที่เราเรียกว่า



แล้วก็ขยายตัวไปสู่วงการธุรกิจการค้าการเงิน

two's complement ซึ่งจะทำให้สะดวกในการนำไปคำนวณเลข two's (หรือ 2's) Complement **นี้คือการเอาค่าเดิมลบออกจากค่ามากที่สุดที่จะเก็บได้นั่นเอง** เช่นในกรณีนี้เรามี 16 บิท ใช้เป็น sign ไป 1 บิท ก็เหลือ 15 บิท จึงเก็บเลขได้สูงสุดถึง 2^{15} = 32,768 เท่านั้น ดังนั้นเลข -1,350 ก็จะเก็บได้เป็น sign bit = 1 และเลขอีก 1 บิทที่เหลือเป็น 32,768-1,350 = 31,418 ซึ่งแปลงเป็นเลขฐานสองได้เป็นดังนี้

sign

1	1	1	1	1	0	1	0	1	0	1	1	1	0	1	0

ไบท์หน้า ไบท์หลัง

นี่ก็เป็นวิธีการเก็บตัวเลขในคอมพิวเตอร์ที่ใช้กันอยู่ ถ้าตัวเลขที่เก็บมีมากกว่านี้ ก็อาจเลื่อน sign bit ไปอยู่ไบท์ถัด ๆ ไปเป็น 4 ไบท์ได้ ส่วนการที่ต้องทำเลขลบให้เป็น 2's Complement ก่อนนั้นก็เพื่อสะดวกในการคำนวณ เพราะ**คอมพิวเตอร์จะทำการลบด้วยการบวก** เช่น ถ้าต้องการเอา 1-1,350 ก็เพียงแต่เอา 1 ตั้งบวกด้วยค่า 2's Complement ของ 1,350 คือ 31,418 ก็จะได้ค่า 31,419 ส่วน sign bit ก็เป็น 1 ตามเดิมเพราะผลลัพธ์ติดลบ ดังนั้นการลบก็จะเป็นดังนี้

sign	เลขฐาน 2		เลขฐานสิบ
0	000 0000	0000 0001	1 +
1	111 1111	1011 1010	-1,350
1	111 1010	1011 1011	-1,349

ซึ่งค่าในรูปเลขฐานสองที่ได้ก็จะเป็นค่าเดียวกับที่เราจะได้จากการทำ 2's Complement ของเลข-1,349 นั่นเอง

Precision

ที่นี้คุณก็อาจจะสงสัยว่า แล้วถ้าเลขที่จะเก็บมีค่ามากมายมหาศาลล่ะจะทำอย่างไร เรื่องนี้เราก็แก้ได้โดยยืมวิธีการเขียนแบบวิทยาศาสตร์ (Scientific notation) มาใช้ เช่น 6,291,456 เขียนได้เป็น 0.629145×10^7 ในคอมพิวเตอร์ก็เช่นเดียวกัน เราจะเขียน 6,291,456 (ซึ่งแปลงเป็นฐานสองได้ 110 0000 0000 0000 0000 0000) ได้เป็น 0.110 0000 $\times 2^{23}$ ได้ แต่เนื่องจากเรารู้แล้วว่า 110 0000 นั้นเป็นเลขหลังจุด ส่วน 23 นั้นเป็นเลขชี้กำลังของ 2 (ซึ่งภาษาคณิตศาสตร์เรียกค่า 0.1100 ว่าเป็น mantissa และ 2^{23} เป็น characteristic ส่วน 23 เป็นค่า exponent) ดังนั้นเราก็จะเก็บเพียงเลข 110 0000 กับ 23 เท่านั้น โดยใช้ที่เพียง 2 ไบท์ (7 บิท สำหรับเลขหลังจุดของ mantissa, 1 บิทเป็น sign และอีก 7 บิท สำหรับ exponent ของ 2 กับ 1 บิทเป็น sign bit ของ exponent ในกรณีที่ยกกำลังติดลบ) ดังนั้น

```
Sign
|0|1 1 0 0 0 0 0.|      |0|0 0 1 0 1 1 1|
```

$+0.1100000 \times 2^{0010111}$

หรือ $+0.11 \times 2^{23}$ ในฐานสอง

ซึ่งเท่ากับ $+0.75 \times 8,388,608$ ในฐานสิบนั่นเอง เพราะ 0.11 ฐานสอง =0.75 ฐานสิบ และ 2^{23} มีค่าเท่ากับ 8,388,608 ซึ่งเมื่อคูณออกมาก็จะได้ 6,291,456 พอดี!

ที่นี้คุณก็เห็นแล้วว่าการใช้ scientific notation ช่วยให้เราเก็บตัวเลขเยอะ ๆ ได้โดยใช้เนื้อที่น้อยกว่าการเก็บด้วยการแปลงเป็นฐานสองธรรมดา (ซึ่งก็คือวิธีการเก็บค่าของตัวแปรที่เป็นแบบ integer ในภาษาคอมพิวเตอร์ทั้งหลายนั่นเอง) แต่การเก็บแบบใหม่นี้ (คือที่ใช้กับตัวแปรแบบ Real ทั้งหลายนั่นเอง) ก็จะนำเราไปสู่ปัญหาต่อไปในเรื่องของความเที่ยงตรง (Precision) เพราะการที่จะแปลงเลขฐานสิบให้อยู่ในรูปฐานสองหลังทศนิยมเพียง 7 บิทได้พอดีดังในตัวอย่างนั้นย่อมทำได้เฉพาะบางจำนวนเท่านั้น บางจำนวนเช่น 6,291,457 จะแปลงเป็นฐานสองได้เป็น 110 0000 0000 0000 0000 0001 ซึ่งต้องใช้ที่หลังจุด (เนื่องจากไม่ใช่เลขฐานสิบ ผมจึงใช้คำว่า "จุด" เฉย ๆ แทน "จุดทศนิยม" ในกรณีนี้บางท่านก็ใช้ว่า "จุดทวินิยม" แทน) ถึง 23 บิทในการเก็บจึงจะได้ค่าที่ถูกต้อง ดังนั้นในทางปฏิบัติจึงยอมให้มีความคลาดเคลื่อนจากการปัดตัวเลขบิทท้าย ๆ ไปบ้าง ซึ่งเรียกว่า round-off error ดังในตัวอย่างนี้ เลขฐานสองทั้ง 23 บิทจะถูกปัดเลข 1 บิทสุดท้ายขึ้นมา 1 หลักก่อนเป็น 110 0000 0000 0000 0000 0010 แต่ก็ยังไม่พอเก็บอยู่ดี ก็เลยถูกตัดเหลือแค่ $0.110\ 000 \times 2^{23}$ เท่ากับค่าที่ได้จากการเก็บ 6,291,456 เหมือนกัน!

ยิ่งไปกว่านั้นเมื่อเลขฐานสิบที่เราใช้มีจุดทศนิยมเข้ามาเกี่ยวข้องเช่น 1.23 เราไม่สามารถจะแปลงค่าดังกล่าวให้เป็นเลขฐานสองได้พอดีไม่ว่าในกรณีใด ๆ เพราะจะกลายเป็นตัวเลขหลังจุดที่ยาวเหยียดไม่รู้จบ ดังนั้นจึงพยายามเพียงให้ได้ค่าใกล้เคียงที่สุดไว้ก็พอ โดยให้ได้ทศนิยมมากตำแหน่งที่สุด คือใช้ที่ให้มากไบท์ขึ้นในการเก็บ mantissa ดังนั้นจึงเกิดมีตัวแปรแบบ Real ที่เป็น single หรือ Double precision ขึ้น โดยถ้าเป็นแบบ Double precision ก็จะใช้จำนวนไบท์มากขึ้นเป็นสองเท่าของธรรมดา เช่นในภาษา BASIC ทั่วไป ตัวแปรแบบ Real เคยใช้ที่เก็บ Mantissa 2 ไบท์ ก็เพิ่มเป็น 4 ไบท์ เพื่อให้ได้ค่าใกล้เคียงกับค่าจริง ๆ มากยิ่งขึ้น เช่น แสดงเลขหลังจุดทศนิยมได้มากขึ้นจาก 7 หลัก เป็น 16 หลัก เป็นต้น

มาถึงตอนนี้ เรื่องราวมันก็มักจะไม่ง่ายอย่างที่ตั้งใจจะเขียนไว้ตอนแรกเสียแล้ว บางท่านที่อ่านมาจนถึงตรงนี้แล้วก็อาจจะเกิดความคิดขึ้นว่า ที่แท้เจ้าคอมพิวเตอร์นี่ก็คิดเลขเร็วเสียเปล่า จะให้ถูกต้องเป็นร้อย ๆ หลักก็ไม่ได้ ถ้าต้องการผลลัพธ์ที่ถูกต้องมากขนาดนั้น ก็คงต้องหาคนที่ทู่ซี้หน่อยมานั่งคิดแทนจะดีกว่า แต่อย่าลืมนะครับว่าคอมพิวเตอร์นี่มันก็เป็นจอมทู่ซี้ยิ่งกว่าคนเสียอีก ถ้าจะให้มันคิดเลขฐานสิบแบบที่มนุษย์เราคิดก็ได้ครับ โดยใช้เลขฐานสิบ 1 หลักต่อหนึ่งไบท์ เอ๊ะ! มากไป เอาแค่ 1 นิบเบิ้ล (4 บิท) ก็พอ ดังนั้นเลขยาว 100 หลักก็ต้องใช้ที่ 50 ไบท์เก็บกันละครับ แล้วก็ให้เครื่องมันทู่ซี้คำนวณ นี่ก็คือวิธีการที่เรียกว่าการคำนวณแบบ packed decimal ในเครื่อง IBM 370 หรือในไมโครก็มีตัว interpreter ภาษา BASIC เช่น PBASIC (Precision BASIC) ที่เขียนโดย Edel Systems และจัดจำหน่ายโดย Lifeboat Association (New York) ที่ให้การคำนวณถูกต้องถึงทศนิยมตำแหน่งที่ 122 และเลขยกกำลัง (characteristic หรือส่วนของ exponential) ได้ถึง 10^{+250} แต่แน่นอนว่าการทำงานก็ต้องช้ากว่าการคำนวณแบบธรรมดาอยู่บ้าง

เท่าที่เล่ามา ก็คงพอเป็นแนวทางให้เห็นการทำงานภายในระบบคอมพิวเตอร์ และลักษณะการแทนข้อมูลแบบต่าง ๆ ภายในเครื่องพอสมควรนะครับ ฉบับหน้าเราจะเปลี่ยนไปคุยเรื่องทางซอฟท์แวร์บ้างนั่นคือ โปรแกรมควบคุมระบบ (Operating System) พบกันใหม่ฉบับหน้า สวัสดีครับ ■

คอมพิวเตอร์กับระบบ ON-LINE

และแล้ว (เมื่อนานมาแล้ว) ก็ถึงยุคของ ON-LINE

แรกเริ่มเดิมทีนั้นการใช้งานคอมพิวเตอร์ ยังจำกัดอยู่เฉพาะงานด้านวิทยาศาสตร์เช่นใช้ในการคำนวณที่ซับซ้อนหรือมีจำนวนมาก ๆ ผู้ใช้คอมพิวเตอร์ก็มีไม่มากนักและคอมพิวเตอร์ในระยะนั้นก็ยังทำงานช้า ดังนั้นผู้ใช้ (USER) จึงสามารถติดต่อกับคอมพิวเตอร์ได้โดยตรง

ต่อมาได้มีการพัฒนาคอมพิวเตอร์ให้เร็วขึ้น และใช้กับงานได้หลายประเภท จำนวนผู้ใช้จึงเพิ่มขึ้นตามความสามารถของคอมพิวเตอร์ การที่จะให้ USER ทุกคนติดต่อกับคอมพิวเตอร์โดยตรงนั้นจึงเป็นไปไม่ได้อีกแล้ว ดังนั้นการประมวลผลโดยใช้คอมพิวเตอร์จึงใช้วิธีการรวบรวมโปรแกรมของหลาย ๆ คนเข้าเป็นกลุ่มเดียว และจะต้องมีสื่อหรือตัวกลาง (media) เพื่อใช้บันทึกโปรแกรมเหล่านี้ไว้ ตัวอย่างเช่นบัตรเจาะรู, เทปแม่เหล็ก, จานแม่เหล็ก เป็นต้น เมื่อรวบรวมโปรแกรมได้จำนวนเหมาะสมแล้วจึงส่งเข้าระบบคอมพิวเตอร์พร้อม ๆ กัน ภายในระบบคอมพิวเตอร์จะมีอุปกรณ์ซึ่งอ่านข้อมูลจากตัวกลางแล้วส่งข้อมูลนั้นไปยังหน่วยประมวลผลกลาง (CPU) หลังจากประมวลผลเสร็จก็จะพิมพ์ผลลัพธ์ลงบนกระดาษ เพื่อแจกจ่ายให้ USER แต่ละคนต่อไป

ระบบดังกล่าวนี้เรียกว่า **การประมวลผลแบบกลุ่ม (batch processing)** ตัวอย่างที่คุ้นเคยกันดีก็คือการเจาะบัตรหรือคีย์ข้อมูลลงเทปเพื่อส่งโปรแกรมตามศูนย์คอมพิวเตอร์ต่าง ๆ ในกรณีนี้ตัวกลางที่ใช้บันทึกโปรแกรมคือบัตรเจาะรูและเทปแม่เหล็ก (ที่เป็นม้วนใหญ่ ๆ) ตามลำดับ

ข้อเสียของการประมวลผลในลักษณะ batch นี้ก็คือ เราต้องเสียเวลานานกว่าจะได้ผลลัพธ์ที่ต้องการ ทั้ง ๆ ที่ความจริงแล้วเครื่องคอมพิวเตอร์ทำงานได้เร็วมาก งานบางอย่างที่ต้องการผลลัพธ์ทันทีทันใดจึงไม่เหมาะสมกับการประมวลผลแบบ batch ตัวอย่างของงานดังกล่าวได้แก่การจองตั๋วเครื่องบินโดยสาร, การฝากและถอนเงิน, การค้นหาข้อมูล เป็นต้น

จากความจำเป็นนี้เอง จึงมีการพัฒนาระบบใหม่ขึ้นมาใช้ให้ USER สามารถติดต่อกับคอมพิวเตอร์ได้โดยตรงโดยไม่ต้องใช้สื่อหรือตัวกลางใด ๆ คอมพิวเตอร์จะรับข้อมูลจาก USER

---

*(บทความจากชมรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย)

โดยตรงแล้วประมวลผล (ซึ่งใช้เวลาสั้นมาก) แล้วจึงส่งผลลัพธ์ให้ USER โดยตรงเช่นเดียวกัน ตัดปัญหาการสิ้นเปลืองเวลา ระหว่างที่โปรแกรมถูกทิ้งไว้เฉย ๆ เพื่อรอการรวบรวมให้ได้จำนวนมากพอก่อนส่งเข้าระบบคอมพิวเตอร์ ระบบที่พัฒนาขึ้นมาใหม่นี้เรียกว่า **ระบบ ON-LINE**

กล่าวได้ว่าลักษณะของระบบ ON-LINE นี้ เหมาะสมกับงานเกือบทุกประเภท ดังนั้นในปัจจุบันจึงมีการใช้งาน ON-LINE กันอย่างกว้างขวาง ทั้งงานราชการและเอกชน ดังเช่น ข่าวล่าสุดในขณะนี้ คือมีการทำคอมพิวเตอร์ระบบ ON-LINE เข้ามาใช้กับระบบงานของศาลและกระทรวงยุติธรรม

อย่างไรก็ตาม การประมวลผลในลักษณะ batch นั้นก็ยังคงมีการใช้งานอยู่ เช่นการทำบัญชีประจำปี, การทำรายการจ่ายเงินเดือนให้แก่พนักงาน เป็นต้น งานเหล่านี้เป็นงานที่ทำเป็นครั้งคราว และยังสามารถทำได้นอกเวลาทำงาน(ของคน) อีกด้วย

ON-LINE คืออะไร

ระบบ ON-LINE คือระบบที่ข้อมูลจากเทอร์มินอลถูกนำผ่านข่ายการติดต่อสื่อสารเข้าสู่ระบบคอมพิวเตอร์โดยทันที และหลังจากประมวลผลข้อมูลนั้น ผลลัพธ์จะถูกส่งจากระบบคอมพิวเตอร์ผ่านข่ายการติดต่อสื่อสารกลับไปสู่เทอร์มินอลโดยทันทีเช่นเดียวกัน ลักษณะการรับส่งข้อมูลเป็นการโต้ตอบกันโดยตรง ระหว่างผู้ใช้เทอร์มินอลกับระบบคอมพิวเตอร์นั่นเอง

จากความหมายของระบบ ON-LINE จะเห็นว่าระบบ ON-LINE ประกอบด้วย 3 ส่วนใหญ่ ๆ ดังนี้

1. *เทอร์มินอล (Terminal)* คือแหล่งที่เกิดของข้อมูลซึ่งอยู่นอกระบบคอมพิวเตอร์ เป็นอุปกรณ์ซึ่ง USER ใช้ติดต่อกับระบบคอมพิวเตอร์

เทอร์มินอลซึ่งคุ้นเคยกันดี ได้แก่ จอภาพ และ keyboard แต่ก็ยังมีเทอร์มินอลในลักษณะอื่นอีก เช่น เครื่องอ่าน/เจาะบัตร, เครื่องฝาก/ถอนเงินอัตโนมัติ ฯลฯ

2. *ข่ายการติดต่อสื่อสาร (Communication network)* ต่อระหว่างคอมพิวเตอร์กับเทอร์มินอลซึ่งพอจะแยกได้เป็น 2 แบบ คือ...

แบบ LOCAL ใช้สำหรับเทอร์มินอลที่อยู่ใกล้ ๆ ระบบคอมพิวเตอร์ สัญญาณที่รับ-ส่งจะมีลักษณะเหมือนกันตลอดทาง การต่อแบบนี้ใช้สายต่อเทอร์มินอลเข้ากับระบบคอมพิวเตอร์โดยตรงเลยเสมือนกับเทอร์มินอลนั้นเป็นอุปกรณ์อินพุท-เอาท์พุท (I/O DEVICE) อย่างหนึ่งของคอมพิวเตอร์

แบบ REMOTE ใช้กับเทอร์มินอลที่อยู่ไกล ๆ การต่อแบบนี้จะต้องมีการเปลี่ยนแปลงสัญญาณที่จะส่งเพื่อให้เหมาะสมกับระบบที่ใช้ในการส่งสัญญาณไกล ๆ และจะต้องมีการแปลงสัญญาณที่รับให้กลับเป็นสัญญาณแบบเดิมจากแหล่งกำเนิดสัญญาด้วย

ข่ายการติดต่อสื่อสารนี้มีองค์ประกอบหลักดังนี้คือ

1. **ช่องทางการส่งผ่านข้อมูล (Data Communication Channels)** หรือสายการติดต่อ (Communication Line) เช่น

- สายโทรศัพท์ เป็นสายการติดต่อที่ใช้กันแพร่หลายที่สุด
- เคเบิล ใช้ฝังใต้ดินหรือใต้สมุทร
- ดาวเทียมสื่อสาร ใช้ติดต่อระยะห่างไกลมาก ๆ มีความเร็วในการส่งข้อมูลสูง
- Fiber optics หรือใยแก้วนำแสงเป็นเทคโนโลยีใหม่ล่าสุดที่ใช้ในการติดต่อสื่อสาร

2. **MODEM (Modulator-Demodulator)** เป็นอุปกรณ์ที่ใช้กับสายการติดต่อบางชนิดเช่นสายโทรศัพท์ เนื่องจากสายการติดต่อประเภทนี้ใช้ส่งสัญญาณอนาลอกเท่านั้นจึงต้องมี MODEM เพื่อเปลี่ยนสัญญาณดิจิตอลจากแหล่งกำเนิดสัญญาณให้เป็นสัญญาณอนาลอกส่งไปตามสาย และเปลี่ยนจากอนาลอกเป็นดิจิตอลตามเดิม เพื่อให้สัญญาณที่จุดหมายปลายทางมีลักษณะเหมือนกับสัญญาณจากแหล่งกำเนิด ดังนั้นจึงต้องมี MODEM ทั้งต้นทางและปลายทาง

สำหรับสายการติดต่อบางชนิดที่สามารถส่งสัญญาณดิจิตอลได้เลยหรือเป็นข่ายการติดต่อสื่อสารระยะใกล้คือแบบ LOCAL ก็ไม่ต้องใช้ MODEM

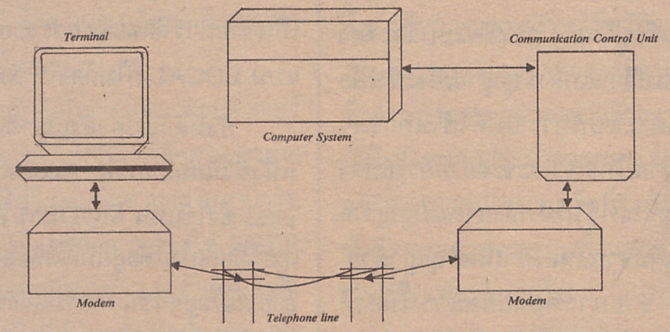
ยังมีตัวแปลงสัญญาณอีกชนิดหนึ่งซึ่งนิยมมากคือ Acoustic coupless ซึ่งใช้แทน MODEM ในกรณีที่เทอร์มินอลไม่ได้อยู่กับที่แน่นอน แต่ต้องย้ายไปเรื่อยๆ เช่น เทอร์มินอลขนาดเล็กของนักข่าวซึ่งตระเวนทำข่าวในที่ต่างๆ รูปร่างของ Acoustic Couplers ก็เหมือนกับหูโทรศัพท์ธรรมดาอันหนึ่ง คงจะเคยเห็นกันบ้างแล้วจากรูปภาพหรือภาพยนตร์ Acoustic Coupler

จะเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณเสียงเพื่อส่งเข้าปากพูดของโทรศัพท และAcoustic Coupler ที่ปลายทางก็จะรับเสียงจากหูฟังเพื่อแปลงกลับเป็นสัญญาณดิจิตอลตามเดิม

3. หน่วยควบคุมการติดต่อสื่อสาร (Communication Control Unit) ต่อระหว่าง MODEM กับระบบคอมพิวเตอร์ ทำหน้าที่ควบคุมการส่งข้อมูล ตรวจจับข้อผิดพลาดในการส่งข้อมูล องค์ประกอบนี้มีขึ้นเพื่อลดภาระของคอมพิวเตอร์ แทนที่จะต้องควบคุมการรับส่งข้อมูลด้วย ก็ทำเพียงการประมวลผลและการควบคุมภายในระบบคอมพิวเตอร์เท่านั้น ทำให้การทำงานเร็วขึ้น

(3) ระบบคอมพิวเตอร์ (Computer System) หมายความถึงอุปกรณ์ต่างๆ (HARDWARE) รวมทั้งโปรแกรมที่เขียนขึ้นเพื่อควบคุมการทำงานของอุปกรณ์ในระบบ (SYSTEM SOFTWARE) ด้วย.

ในระบบคอมพิวเตอร์หนึ่งๆ อาจจะมีหน่วยประมวลผล (processor) มากกว่า 1 หน่วยก็ได้ ซึ่งใช้ทำงานร่วมกันในงานซับซ้อน หรือ ทำงานเป็นอิสระต่อกัน แต่มีไว้เป็นหน่วยสำรองในกรณีที่หน่วยที่ใช้ทำงานหลักเกิดขัดข้องก็ได้ สำหรับระบบ ON-LINE ที่ใช้กับงานง่ายๆ ไม่ซับซ้อนจะมีหน่วยประมวลผลเพียงหน่วยเดียว และเราอาจจะเขียนโปรแกรมให้มันทำหน้าที่ควบคุมการติดต่อสื่อสารด้วยก็ได้ จึงไม่ต้องมีหน่วยควบคุมการติดต่อสื่อสาร เป็นการลดต้นทุนและค่าใช้จ่ายลง แต่การทำงานก็จะมีประสิทธิภาพน้อยกว่า ดังนั้นการพิจารณาเพื่อติดตั้งระบบคอมพิวเตอร์ที่ใช้งาน ON-LINE นั้นจึงขึ้นอยู่กับลักษณะงานที่จะทำเป็นหลักสำคัญ



การทำงานของระบบ ON-LINE

เพื่อให้เข้าใจได้ดี โปรดดูรูปประกอบไปพร้อมๆ กับคำอธิบาย

เริ่มแรก USER จะสั่งข้อมูลเข้าทาง TERMINAL สัญญาณดิจิตอลจะถูกผลิตออกจาก Terminal แล้วถูกส่งไปยัง Modem แล้ว Modem จึงแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกส่งไปตามสายโทรศัพท์ แล้วเมื่อถึงปลายทาง Modem อีกตัวหนึ่งจะเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอลตามเดิมแล้ว Communication Control Unit จะตรวจเช็คข้อผิดพลาดของข้อมูลก่อนจะส่งสัญญาณนั้นเข้า Computer System เพื่อประมวลผลต่อไป

หลังจากได้ผลลัพธ์แล้ว ผลลัพธ์จะถูกส่งกลับไปยัง Terminal ในลักษณะเดิมดังที่อธิบายมา

หมายเหตุ *ตามความเป็นจริงนั้นระบบคอมพิวเตอร์ระบบหนึ่งไม่ได้ต่อกับ Terminal เพียงแห่งเดียวเหมือนกับที่อธิบาย แต่จะต่อไปยัง Terminal มากมายหลายแห่ง ดังนั้นจึงจะต้องมีอุปกรณ์พิเศษเพื่อเป็นตัวเลือกว่าจะให้ระบบคอมพิวเตอร์ติดต่อกับ Terminal ใดในเวลาหนึ่ง รวมทั้งต้องมีระบบการต่อแบบต่างๆ และการควบคุมการทำงานให้เป็นไปด้วยดี เช่นระบบ Time-sharing และ Virtual Machine เป็นต้น ซึ่งจะได้นำมาเล่าสู่กันฟังในโอกาสต่อไป* ■

ผ่า IBM® PC



ตอนที่ 2
«แต่งเครื่อง»

"เรื่องของการ์ดต่าง ๆ ที่ใช้ในการเพิ่มประสิทธิภาพให้กับ IBM PC และ XT"

IBM PC นั้นถูกผลิตออกมาจากโรงงานเหมือน ๆ กัน แต่ก็ไม่ได้หมายความว่าแต่ละเครื่องที่ใช้งานกันอยู่ทั่วไปนี้จะเหมือนกันทีเดียว ทั้งนี้โดยการเสียบอร์ดหรือการ์ดต่าง ๆ ลงไปในช่อง Slot ที่ว่างอยู่จะสามารถทำให้ PC แต่ละเครื่องมีคุณสมบัติที่แตกต่างกันได้ โดยสามารถดัดแปลงให้ตรงต่อความต้องการเฉพาะของผู้ใช้แต่ละคน

บทความนี้จะกล่าวถึงเรื่องราวโดยทั่วไปของฮาร์ดแวร์ต่าง ๆ ที่เป็นที่นิยมนำมาใส่ลงไปในเครื่อง IBM PC และ XT ของคุณ

ก่อนที่เราจะเริ่มลองมาดูบอร์ดที่มาพร้อมกับเครื่องกันเสียก่อน System board นั้นจะติดมาพร้อมกับเครื่องนิยมเรียกกันว่าบอร์ดแม่ (mother board) ซึ่งภายในจะบรรจุไมโครเซสเซอร์ และวงจรต่าง ๆ รวมทั้ง ROM ที่บรรจุโปรแกรมควบคุมระบบงานของตัว PC เองด้วย

ในโมเดลแรก ๆ ของ PC นั้นจะสามารถบรรจุ RAM ได้เพียง 64K เท่านั้นรุ่นดังกล่าวนี้ คือรุ่นที่ที่ผลิตก่อนเดือนมีนาคม 1983 ดังนั้นถ้าจะต้องการขยายหน่วยที่มากกว่านี้จำเป็นจะต้องใช้ Expansion Card มาเสียบ



รูปที่ 1 PC Intellicom ของบริษัท Bizcomp

RAM Card

Ram Expansion Card ที่ผลิตโดยผู้ผลิตรายอื่นที่มิใช่ IBM มีอยู่หลายแบบด้วยกันแบบหนึ่งมีชื่อว่า 512K RAM Card ผลิตโดย Apparat Inc. ใน Card นี้สามารถเสียบชิพได้สูงสุดถึง 512 K (แต่อาจเสียบไม่ถึงก็ได้) นับว่าเป็นบอร์ดที่ราคาไม่แพงนัก ราคาเริ่มตั้งแต่ $ 189 สำหรับ 64 K และ $ 381 สำหรับ 256 K เป็นต้น RAM Card นี้จะมาพร้อมกับ Sdrive ซึ่งเป็นโปรแกรมภาษาเครื่องที่สามารถจะกันเนื้อที่ส่วนหนึ่งในหน่วยความจำสำหรับทำ RAM based disk drive หรือ disk drive ที่สร้างขึ้นจาก RAM โดย Load ข้อมูลเข้ามาเก็บไว้

เมื่อโปรแกรมเช่น wordstar ถูก load เข้าไปใน Sdrive คอมพิวเตอร์จะถือเสมือนว่ามันเป็น disk drive C หรือ D ซึ่ง-Sdrive นี้จะช่วยเพิ่มความเร็วของโปรแกรมในการเข้าถึงและดึงข้อมูลเข้าหรือออกให้มากกว่าการใช้ disk drive ธรรมดานับสิบเท่า

นอกจากนั้น RAM Card ยังสามารถใช้ร่วมกันกับ Combo II Card ซึ่งเป็น multifunction board ของ Apparat เช่นกันโดยนำมาต่อเข้าด้วยกันและเสียบลงบน Slot เพียงช่องเดียวเท่านั้นได้อีกด้วย

Combo II Card เป็นการ์ดที่รวมเอา parallel-printer adapter, RS-232 Cport Clock & Calendar และ game adapter ไว้ด้วยกัน การ์ดนี้ไม่เพียงแต่ Compatible กับ Apple และ IBM joystick เท่านั้นยังสามารถใช้ต่อกับ Mouse ได้อีกด้วย

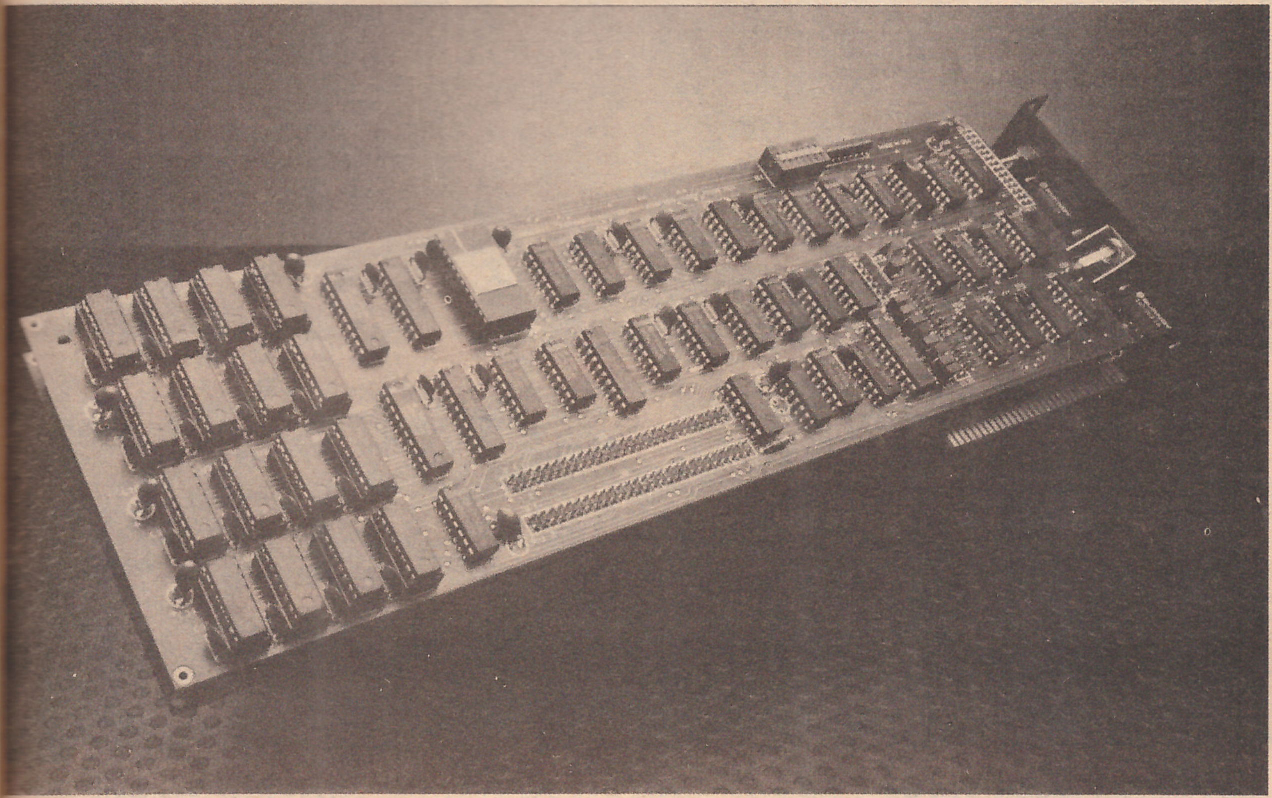
Six Pak Plus

Expansion card สำหรับ PC ที่เป็นที่นิยมที่สุด ดูเหมือนว่าจะเป็นของ บริษัท AST Research ซึ่งมียอดขายถึงประมาณอาทิตย์ละ 1000 บอร์ด การ์ด 256 K ของ Six Pak นั้นมีฟังชั่นได้เช่น เดียวกับ RAM และ Combo Card รวมกันรวมทั้งมี Software ชื่อ Superdrive (เหมือนกับ Sdrive) นอกจากนั้นยังมี Super spool ซึ่งจะทำหน้าที่ใช้ส่วนหนึ่งของ RAM เป็น print buffer ด้วยโปรแกรม-Super spool คุณสามารถใช้งานเครื่องพิมพ์พร้อม ๆ กับทำงานอย่างอื่นได้ด้วย การ์ด 256K Six Pak Plus นี้รวมทั้งซอฟท์แวร์ขายในราคา $ 695 และเพิ่ม $ 100 ทุกๆ 64 K

Six Pak Plus สามารถใช้งานได้กับ System board แบบเก่าคือ 64 K และ System board แบบใหม่คือ 256 K ด้วย SIX Pak Plus มี RAM board ชนิด 384 K เพื่อใช้กับ board แบบใหม่เพื่อให้ได้หน่วยความจำเต็มที่ของ PC นั่นคือ 640K

Bus Board

ยังมีบอร์ดอีกชนิดหนึ่งที่ผู้ใช้สามารถจะเพิ่มฟังชันต่าง ๆ ลงไปในบอร์ดได้เอง บอร์ดนี้มีชื่อว่า Busboard จากบริษัท LNW Computer Corp มันเป็นแผ่น master board ที่ใช้เสียบลงบน Slot เดียวของ IBM PC บอร์ดนี้จะมี Socket สำหรับเสียบ Module ของฟังชั่นต่าง ๆ



รูปที่ 2 Multidisplay Card จาก Paradise Systems

ที่คุณต้องการอยู่ทั้ง 2 ด้านของบอร์ด และยังมีที่เสียบชิพหน่วยความจำได้มากถึง 512 K อีกด้วย Module ที่ใช้เสียบได้มีอาทิเช่น clock/calendar ซึ่งตั้งเวลาได้ถึง 10 ค่ารวมทั้งซอฟท์แวร์ชื่อ "Bus-time" ซึ่งสามารถโปรแกรมคอมพิวเตอร์ของคุณทำงานในเวลาที่คุณไม่อยู่ได้ asynchronous communication module ซึ่งใช้แทน IBM RS-232C card, parallel printer module, game controller module, 300 bit/sec auto dial modem module และในอนาคต LNW ได้วางแผนจะออก Z80 A coprocessor, 6502 coprocessor และ graphic controller module ซึ่งให้ resolution ขนาด 1024 × 1024 และ 16 สี ออกมาด้วย

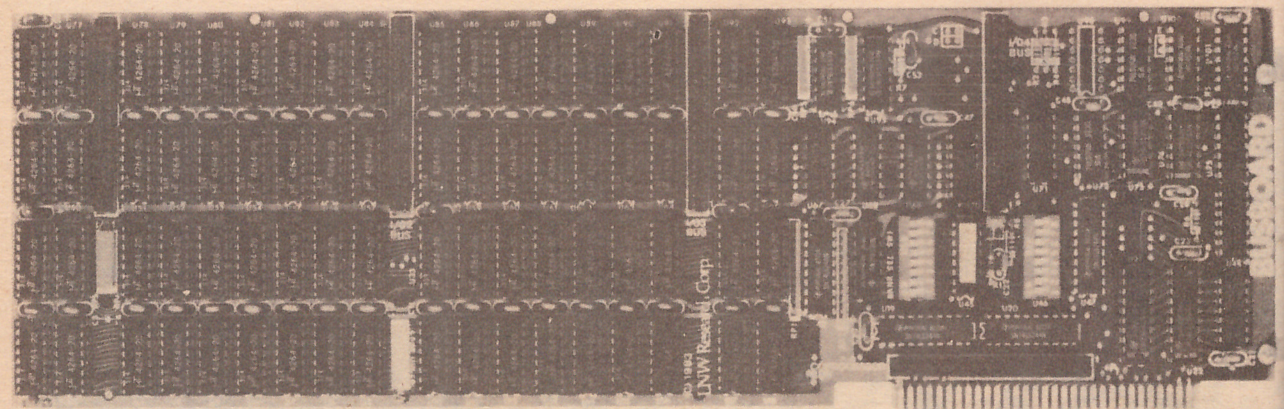
Busboard ขายมาพร้อมกับซอฟท์ แวร์เช่นเดียวกับ board ทั่ว ๆ ไป โดยมีโปรแกรม Spoolbus ซึ่งไม่เพียงแต่เป็น printing buffer เท่านั้นยังสามารถจัดเรียงไฟล์ที่จะพิมพ์ในเครื่องพิมพ์ถึง 5 ไฟล์ด้วยกัน และมี Busdrive ซึ่งจะสร้าง RAM drive ขึ้นในหน่วยความจำและทำงานได้เร็วกว่า disk drive ธรรมดารวมทั้ง hard disk ด้วย

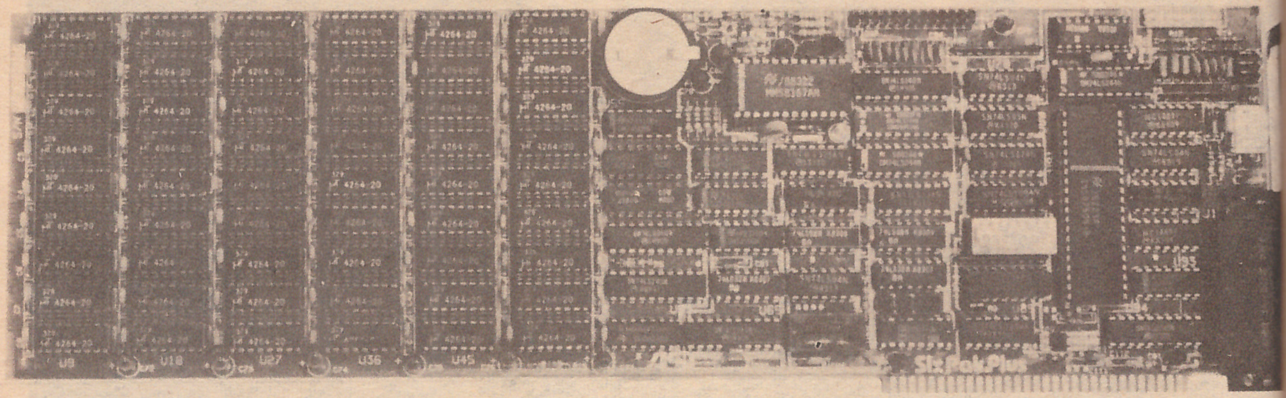
ถึงแม้ว่า Busboard จะดูเหมือนว่าเป็นบอร์ดสารพัดประโยชน์ที่แท้จริง แต่มันก็ยังมีราคาที่สูงพอสมควรทีเดียว บอร์ดที่มี RAM 256 K และมี Module clock, serial และ parallel prot และ game I/O confrol ราคาสูงถึง $900 ทั้งการติดตั้งบอร์ดก็ทำไม่ง่ายเช่นกับบอร์ดอื่น อาจจะต้องใช้ช่างเทคนิคที่มีความชำนาญช่วยทำให้

บอร์ดควบคุม Output

มีบอร์ดอีกประเภทหนึ่งซึ่งมักจะใช้ควบคุม output จากระบบเช่นคุมจอภาพหรือเครื่องพิมพ์ ซึ่งก็มีบริษัทอื่นนอกเหนือจาก IBM สร้างบอร์ดที่มีคุณสมบัติน่าสนใจออกมาจำหน่ายเหมือนกัน ก่อนจะดูกันถึงบอร์ดเหล่านี้ลงมาดูกันก่อนว่า IBM มีบอร์ดอะไรให้บ้าง

การออกแบบ PC ของ IBM อย่างหนึ่งที่เป็นที่วิพากวิจารณ์กันมาก ก็คือการออกแบบการออกจอที่ไม่เหมือนคอมพิวเตอร์อื่น ๆ ที่เป็นที่นิยมกัน เช่น Apple อย่างใน Apple II นั้น monitor สามารถจะแสดงตัวหนังสือและภาพกราฟิคได้ทันทีซึ่งนับว่าเป็นการสะดวกสำหรับผู้ใช้ที่เมื่อซื้อจอภาพมาก็ต่อใช้งานทุกประเภทได้ แต่ IBM ไม่เป็นอย่างนั้น คุณจำเป็นต้องซื้อบอร์ด





รูปที่ 3 BUS board สามารถเพิ่ม RAM ได้ถึง 512 K

รูปที่ 4 Six Pak Plus สามารถเพิ่ม RAM ได้ 384 K.

monochrome display/parallel-printer adapter พร้อมกับจอขาวดำหรือ Color graphic adapter พร้อมกับจอสี

monochrome นั้นเหมาะสำหรับงานด้าน word processing ซึ่งให้ภาพตัวอักษร 80 ตัว x 25 บรรทัดที่มีความคมชัดดีมาก แต่ไม่สามารถเล่นเกมส์ได้ เพราะมักจะต้องการจอกราฟิคสีแทน

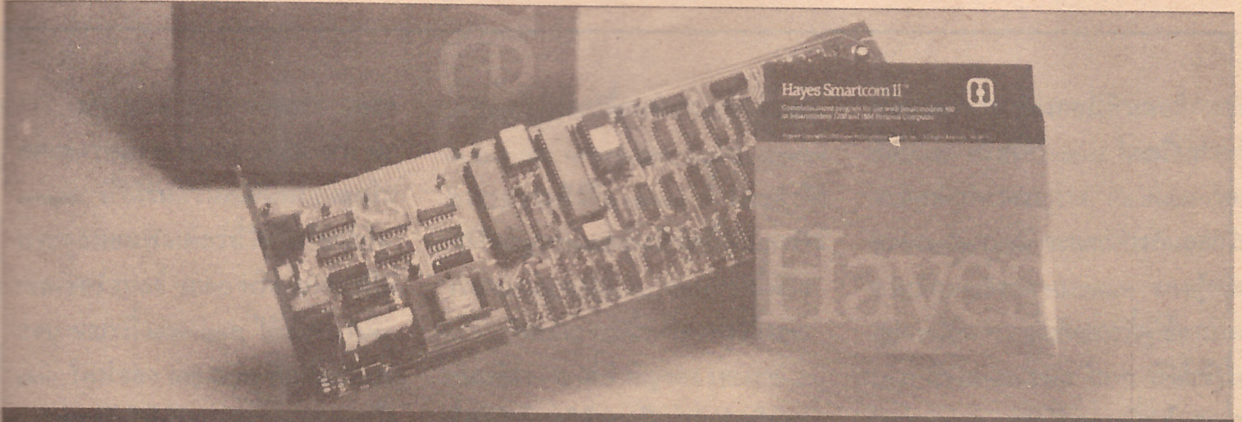
ในขณะเดียวกันถ้าคุณไปซื้อจอสีและ color-graphic adapter มาเวลาคุณนั่งลงใช้เครื่องเขียนรายงานสักฉบับหนึ่ง คุณจะเห็นตัวอักษรที่ไม่ค่อยคมชัดนักอ่านนาน ๆ ก็อาจปวดตาได้ (จอสีมีความละเอียดประมาณ 128,000 pixels ในขณะที่จอขาวดำมี ถึง 252,000 pixels)

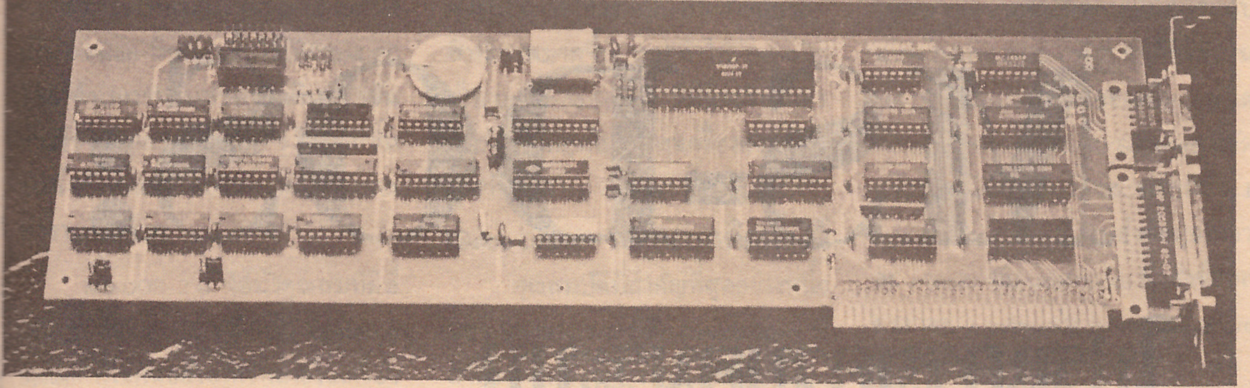
ในการที่จะใช้งานทั้ง 2 ประเภทให้ดีเท่า ๆ กันนั้นคุณต้องซื้อจอทั้ง 2 จอ รวมทั้ง Adapter ทั้ง 2 แบบด้วย ซึ่งคุณจะต้องเสียเงินรวมกันแล้วพอกับราคาเครื่อง Apple II เครื่องหนึ่งทีเดียว มีทางไหนไหมที่จะแก้ปัญหาของ IBM ที่ว่านี้ได้ ที่จริงแล้วมีหลายวิธีด้วยกัน วิธีแรกคุณสามารถรวมการ์ดทั้ง 2 ไว้ด้วยกันโดยการซื้อ Multidisplay Card จากบริษัท Paradise System ซึ่งมีสามารถใช้งานกับจอของ IBM ได้ทุกแบบ ไม่ว่าจะเป็น monochrome หรือจอสีไม่ว่าจะเป็นสัญญาณแบบ Composite Video หรือ RGB และยังมี parallel printer ได้อีกด้วย การ์ดนี้ราคา $ 595 ในกรณีที่คุณใช้จอสีและขาวดำคุณสามารถใช้สวิทช์เปลี่ยนกลับไปกลับมาระหว่างจอทั้ง 2 ได้หรือใช้คำสั่ง DOS ในการควบคุมก็ได้ ซึ่งนอกจากจะสะดวกในการใช้งานแล้วยังเป็นการประหยัด slot ไปอีก 1 ช่องอีกด้วย

High-Resolution Graphic

บริษัท Amdek Corp. ได้เสนอทางเลือกอีกแบบหนึ่งของปัญหาข้างต้น โดยได้เสนอ MAI (Multiple-Adapter Interface) Board ซึ่งเป็น Card high resolution สำหรับจอสีโดยมีความละเอียดได้สูงสุด 640x400 จุดซึ่งเท่ากับจอขาวดำและยังแสดงสีได้ 4 สี ซึ่งเท่ากับประหยัดจอภาพขาวดำไป 1 จอทีเดียว กราฟิคใน Mode อื่น ๆ ก็มีอาทิเช่น 320x200 ใน 16 สี, 640x200 ใน 4 สี, 320x400 ใน 16 สี และยังมี Parallel printer Interface ให้อีกด้วย

MAI board มีซอฟท์แวร์มาให้ด้วยชื่อว่า "Halo" ซึ่งเป็นโปรแกรมช่วยเขียนภาพกราฟิคซึ่งมีคำสั่งกว่า 100 คำสั่งที่ใช้ได้ใน High Resolution mode สำหรับการสร้างเส้นตรง, เส้นโครง, วงกลม,





รูปที่ 5 Combo II Card ใส่ RAM ได้ 512K

รูปที่ 6 Smart modem 1200 B ของ Hayes

วงรี, สี่เหลี่ยมและรูปทรงอื่น ๆ

นอกจากนั้น MAI board ยังมี RAM 128 K ซึ่งถ้าทำงานใน IBM Mode ธรรมดาแล้วสามารถนำเนื้อที่ใน RAM ที่ประมาณครึ่งหนึ่งไปใช้งานอย่างอื่นได้รวมทั้งอีก 32K ไปใช้เป็น print spooler สำหรับเก็บ file ที่คอยจะพิมพ์ออกทาง printer ได้อีกด้วย ใน high resolution mode นั้น MAI จะใช้ RAM ทั้ง 128K เป็น display RAM โดยคุณสามารถที่จะเก็บ Graphic screen ไว้ใน RAM ได้ถึง 8 Screen โดยไม่ต้องเก็บไว้ใน disk เลย และที่พิเศษก็คือในบอร์ดมีวงจรและอินเตอร์เฟสสำหรับ light-pen สำหรับในการใช้เลือกเมนู, Graphic editor และวาดภาพบนจอ คุณสมบัติต่าง ๆ เหล่านี้ทำให้ MAI board เป็น high resolution graphic board ที่น่าสนใจที่สุดในตลาดขณะนี้

Communication board

บอร์ดนี้มักจะเป็นบอร์ดที่ผู้ใช้เลือกซื้อเมื่อได้เลือกซื้อบอร์ดประเภทอื่น ๆ เช่น RAM, spooling, Graphic และ interface ต่าง ๆ จนครบแล้ว

แต่เดิมนั้น IBM PC ติดต่อกับระบบคอมพิวเตอร์อื่น ๆ ได้โดยผ่านทาง Modem โดยใช้สาย RS-232 C ซึ่งนอกจากจะต้องเสียเงินซื้อตัว Modem แล้วยังต้องซื้อสายเคเบิลในการเชื่อมต่อกับ IBM PC และยังต้องซื้อ Communication Adapter อีกด้วย

เดี๋ยวนี้ได้มีผู้คิดจะลดความยุ่งยากและค่าใช้จ่ายเหล่านี้ลงโดยได้รวมทุกอย่างไว้ ในบอร์ดแผ่นเดียวโดยเสียบลงใน Slot ของ PC เลย ทำให้ไม่เปลืองเนื้อที่ ไม่ต้องมี Communication adapter ไม่ต้องมีสายเคเบิ้ล และ power supply พิเศษ แต่อย่างใด board ดังกล่าวมีให้เลือกตั้งแต่ความเร็วในการส่ง 300 ถึง 1200 บิทต่อวินาที (Baud)

บอร์ดที่กำลังเป็นที่นิยมอยู่ในขณะนี้ได้แก่ Smart modem 1200 B จาก บริษัท Hayes Microcomputer Product ซอฟท์แวร์ที่มากับบอร์ดนี้มีชื่อว่า Smartcom II จะช่วยเหลือผู้ใช้ในการจัดการระบบสื่อสารต่าง ๆ ที่สามารถติดต่อกับไมโครคอมพิวเตอร์หรือคอมพิวเตอร์ระบบอื่น ๆ ได้ทุกระบบ

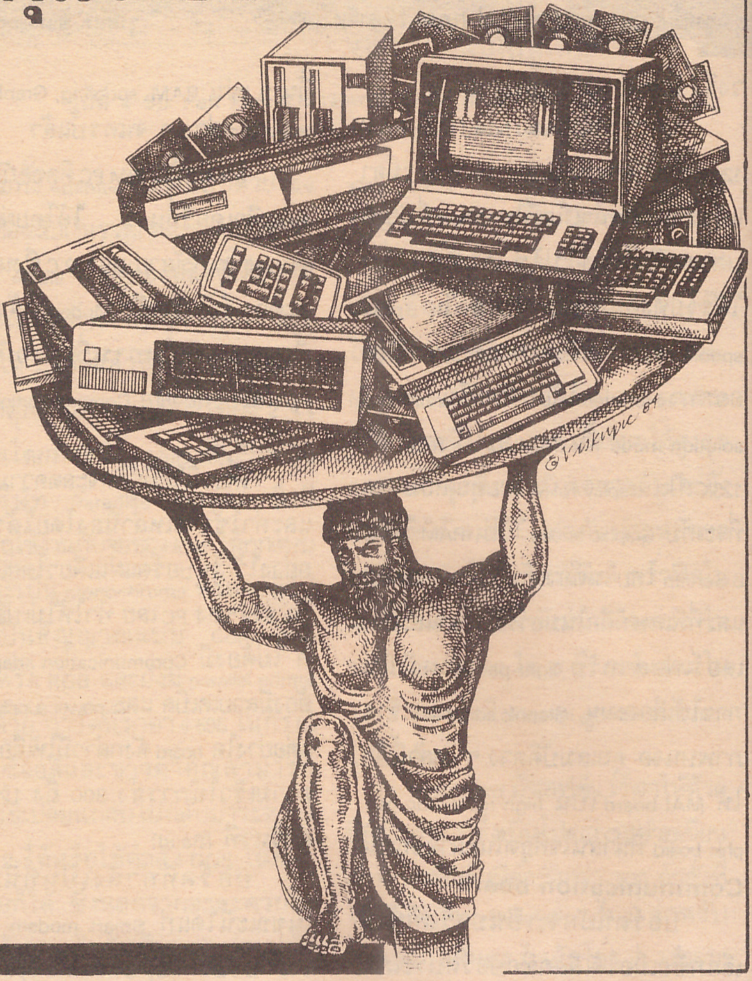
นอกจาก Smartmodem แล้วยังมี communication board อื่น ๆ อีกที่คุณสมบัติคล้าย ๆ กับ Smartmodem อาทิเช่น PC: Intellimodem, Crosstalk XVI และ PC Talk III ซึ่งต่างก็มี ราคาและคุณสมบัติปลีกย่อยที่แตกต่างกันไปบ้าง

คงจะเห็นได้ชัดเจนขึ้นแล้วว่าตลาด expansion-board ของ IBM PC นั้นได้มีส่วนอย่างสำคัญในการพัฒนาระบบไมโครคอมพิวเตอร์ บอร์ดต่างๆที่นำมาเสนอเป็นแต่เพียงส่วนน้อยเท่านั้นในตลาดที่จะมีส่วนช่วยให้คอมพิวเตอร์ของคุณมีพลังและประสิทธิภาพมากขึ้นรวมทั้งทำให้คุณสามารถที่จะเสริมแต่งเครื่อง IBM PC ของคุณให้เหมาะกับงานของคุณได้ในที่สุด ■

MS-DOS:

โปรแกรมควบคุมระบบ ยุค 16 บิท

ตอนที่ 1



สุมคิด บุญประภากร

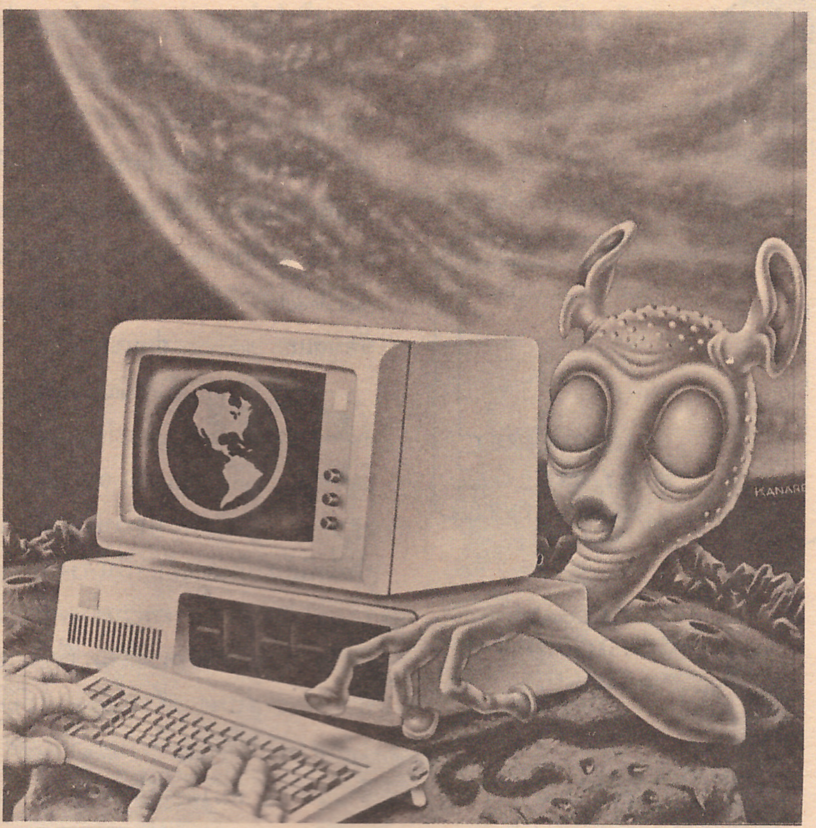
MS-DOS เป็น SINGLE USER OPERATING SYSTEM ของบริษัท Microsoft ที่ได้รับความนิยมมากที่สุดในตระกูลคอมพิวเตอร์ที่ใช้ไมโครโปรเซสเซอร์ เบอร์ 8086 หรือ 8088 เป็น CPU ขณะนี้ Microsoft ได้พัฒนา version ใหม่ของ MS-DOS คือ MS-DOS 2.0 และยังได้พัฒนา PC-DOS 2.0 ซึ่งเป็น version พิเศษ เพื่อใช้กับ IBM PC โดยได้ทำการปรับปรุงเพิ่มเติมฟังก์ชั่นการใช้งานและคำสั่งต่าง ๆ ให้มากขึ้น อีกทั้งยังพัฒนาให้เพิ่มความเร็วในการทำงานขึ้นด้วย

MS-DOS 2.0 ได้นำคุณสมบัติที่เด่น ๆ ของ UNIX (ซึ่งเป็น Operating System ที่ใช้กันอย่างแพร่หลายในเครื่อง mainframe และมินิคอมพิวเตอร์) เช่น ระบบ file ตามลำดับชั้น (Hierarchical file System), การเปลี่ยนทิศทางของ Input Output (I/O redirection), piping, spooler มารวบรวมไว้ใน version นี้ และยังได้แก้ไขปัญหาบางอย่างที่เกิดขึ้นในการใช้งาน MS-DOS version ก่อน ๆ นี้ด้วย

ในบทความนี้จะได้กล่าวถึงคุณสมบัติต่าง ๆ ที่สำคัญของ MS-DOS 2.0 ซึ่งทำให้การใช้งานเป็นไปอย่างมีประสิทธิภาพมากขึ้น และทำให้ MS-DOS เป็น Operating System ที่มีชื่อเสียงและแพร่หลายที่สุดในบรรดาเครื่องคอมพิวเตอร์ขนาด 16 bit ขณะนี้

Divice Driver ที่สามารถติดตั้งได้

คุณสมบัติที่สำคัญที่สุดอย่างหนึ่งของ MS-DOS 2.0 ก็คือ Device Driver ที่สามารถติดตั้งอยู่ใน Operating System ได้ (Device Driver นั้นหมายถึง Software routine ซึ่งใช้ในการควบคุมอุปกรณ์รับส่งข้อมูลต่าง ๆ ของคอมพิวเตอร์ เช่น คีย์บอร์ด หรือจอ monitor โดยเป็นส่วนที่จะติดต่อกับวงจรทาง Hardware โดยตรง Device Driver จะแตกต่างกันออกไปตามลักษณะทาง Hardware ของผู้ผลิตอุปกรณ์นั้น ๆ) นั่นคือการที่ผู้ใช้สามารถ Load ตัว device driver ใด ๆ ก็ได้เข้าไปในหน่วยความจำในตอนเริ่มต้นระบบตามต้องการ device ต่าง ๆ ที่ใช้กันใน MS-DOS สามารถแบ่งออกเป็น 2 ชนิด คือ block device ซึ่งเป็น Software ที่จัดการรับส่งข้อมูลคราวละมากกว่า 1 byte เช่น disk drive ซึ่งรับส่งข้อมูลคราวละ 1 sector กับ serial device ซึ่งส่งข้อมูลคราวละ 1 byte เช่น Printer, Keyboard หรือจอภาพ



แต่เดิมนั้นเป็นความรับผิดชอบของบริษัทผู้ผลิตอุปกรณ์ Hardware ที่ใช้กับคอมพิวเตอร์ที่จะต้องจัดหา BIOS (Basic Input/Output System) ที่จะนำมาใช้กับ Operating System หนึ่ง ๆ เพื่อให้สามารถ run ในสภาพแวดล้อมของ Hardware ของบริษัทนั้น ๆ ได้ ซึ่งทำให้เกิดปัญหาขึ้นในกรณีที่ Operating System นั้นเปลี่ยน BIOS version ใหม่ ยกตัวอย่าง เช่น บริษัทต่าง ๆ ซึ่งผลิต Hard disk สำหรับใช้กับเครื่อง IBM PC จะต้องเขียน BIOS ส่วนที่ติดต่อกับ Hard disk ของบริษัทตนมารวมไว้ใน BIOS ของ IBM PC เอง (เพราะ IBM PC มี BIOS ที่ติดต่อกับฮาร์ดดิสค์ของ PC เองอยู่แล้ว จึงจะสามารถขายฮาร์ดดิสค์ได้ โดยต้องให้ BIOS ตัวใหม่นี้กับผู้ใช้ด้วย) ทำให้เกิดสภาวะไม่เป็นมาตรฐานของ Software ขึ้นเพราะเมื่อ IBM PC เปลี่ยน BIOS version ใหม่ครั้งหนึ่งทุกๆบริษัทจะต้องเขียน BIOS ตัวใหม่ทุกครั้ง

MS-DOS 2.0 ได้แก้ปัญหานี้โดย

เพิ่มส่วนที่ทำให้ผู้ใช้สามารถติดตั้ง divice Driver ของตนเองขึ้น จึงเป็นการลดภาระของผู้ผลิตอุปกรณ์คอมพิวเตอร์ลง เช่น ผู้ผลิต Hard disk เพียงแต่เขียน Hard disk device driver ให้เฉพาะกับผู้ใช้ที่ซื้อ Hard disk ของบริษัทตนเท่านั้น ส่วนผู้ใช้ที่ไม่มี Hard disk ก็จะประหยัดเนื้อที่หน่วยความจำที่จะใช้เก็บ Operating System ลงเพราะไม่ได้ติดตั้ง Hard disk device driver เอาไว้

อีกลักษณะหนึ่งที่แสดงถึงประโยชน์ของการติดตั้ง driver ได้ก็คือการใช้ Keyboard ภาษาต่างประเทศ เช่น ผู้ใช้สามารถจัดระบบของ Keyboard ใหม่จากภาษาอังกฤษ เป็นภาษาฝรั่งเศส โดยติดตั้ง French Keyboard driver เข้าไปแทน driver ตัวเดิม Keyboard ซึ่งเดิมเป็นภาษาอังกฤษก็จะทำงานเหมือนถูกออกแบบมาเพื่อใช้ในโปรแกรมที่เป็นภาษาฝรั่งเศส

ยิ่งไปกว่านั้น MS-DOS 2.0 สามารถติดตั้ง divice driver แบบ serial stream processing ซึ่ง Microsoft ได้เป็นตัวแทนในการกำหนดมาตรฐานของ driver ชนิดนี้ 2 ตัว คือ ANSI (American National Standards Institute) terminal driver standard of cursor opsitioning ซึ่งจะใช้เป็นมาตรฐานของการกำหนดตำแหน่งของ cursor ใน terminal ทั่ว ๆ ไปและ Virtual Device Interface (VDI) graphic serial stream interpreter ซึ่งจะกำหนดขึ้นเพื่อใช้เป็นมาตรฐานของคำสั่ง vector operation พื้นฐานในระบบ graphic output device

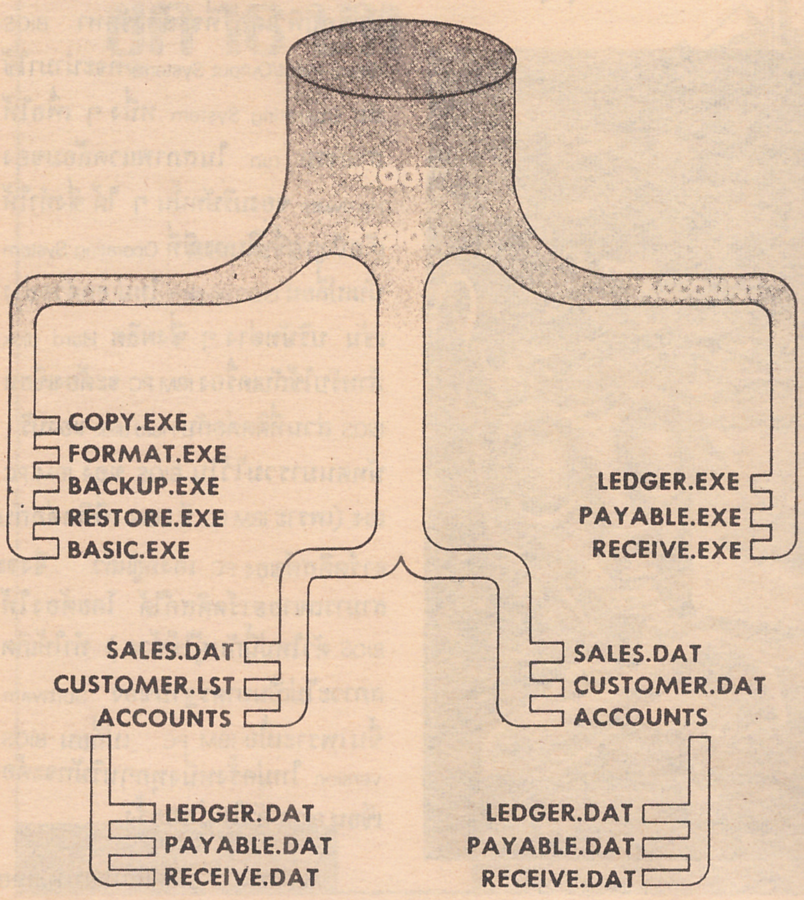
ระบบ file ตามลำดับชั้น (Hierarchical file system) ของ MS DOS 2.0

โครงสร้าง file ตามลำดับชั้นของ MS-DOS ได้นำมาจาก Xenix (ซึ่งเป็น Operating System ที่ Microsoft ดัดแปลงมาจาก UNIX) โดยเป็นระบบที่จัดกลุ่มของ file เป็นชั้น ๆ เพื่อความสะดวกในการใช้งาน ข้อดีของระบบนี้คือ เราสามารถแบ่งเนื้อที่ใน Disk ออกเป็นส่วน ๆ ให้กับผู้ใช้หลาย ๆ คน ผู้ใช้แต่ละคนก็จะมี directory ย่อยของตนเอง และ access file เฉพาะใน directory ของตน การค้นหา file การเก็บรักษาและการนำ file มาใช้งานก็ทำได้สะดวกยิ่งขึ้น โดยเฉพาะในระบบที่ใช้ Hard disk เป็นลำดับขั้นไม่เพียงแต่จะลดเวลาการค้นหา file เท่านั้น แต่ยังจะช่วยให้สามารถเก็บ file ของผู้ใช้แต่ละคนไว้รวมกันโดยไม่ปะปนกับ file ของผู้อื่น

ลักษณะโครงสร้างของ file

โครงสร้าง file ของ MS-DOS จะมี directory หลักเรียกว่า root directory ซึ่งสามารถเก็บได้ทั้ง file ข้อมูล (data file) ธรรมดา และ directory ย่อย (subdirectory) ซึ่งเป็น file ชนิดพิเศษ โดยแทนที่จะเก็บข้อมูลตามปกติ แต่ directory ย่อยเหล่านี้ จะเก็บ file ต่าง ๆ เหมือนกับ root directory คือสามารถเก็บได้ทั้ง file ข้อมูลและ directory ย่อยของมัน

รูปที่ 1 แสดงถึงลักษณะของระบบ file หลายระดับ โดย disk แต่ละแผ่นจะมีรากหรือ "root" ของมันเอง (ถ้าเป็นระบบที่มีหลาย drive ก็จะเปรียบได้กับป่าซึ่งมีต้นไม้หลายต้น) ในทางทฤษฎีจะไม่มีข้อจำกัดเรื่องจำนวนชั้นของ file (หรือความซับซ้อนของต้นไม้) ซึ่งหมายความว่าเราสามารถสร้าง Directory ย่อยได้ลึกลงไปได้เรื่อย ๆ ข้อจำกัดของจำนวน file ที่มีได้อยู่ที่เนื้อที่ของ disk (หรือ Hard disk) เท่านั้น



การเรียกชื่อ file ในระบบนี้จะเรียกโดยใช้ "path name" ซึ่งเป็นรายชื่อของ directory ย่อยและชื่อของ file (ที่ถูกเก็บอยู่ใน directory ย่อยเหล่านั้น) นำมาประกอบกันเข้าเพื่อใช้อ้างถึง file หนึ่ง ๆ ในทุกระดับของระบบนี้ ชื่อแต่ละชื่อจะถูกคั่นด้วย Back Slash (\) ตัวอย่างการอ้างถึง file ในรูปที่ 1 เช่น

\SYSTEM\BASIC.EXE

\REGION\WEST\SALES.DAT

\REGION\EAST\ACCOUNTS

\LEDGER.DAT

ในตัวอย่างสุดท้ายเป็นการอ้างถึง file ชื่อ LEDGER.DAT ซึ่งอยู่ภายใน directory ย่อยชื่อ ACCOUNTS โดย ACCOUNTS อยู่ใน directory ย่อยชื่อ EAST ก็อยู่ใน directory ย่อยชื่อ REGION ซึ่งเป็น directory ย่อยของ root directory ในแผ่น disk

MS-DOS มีคำสั่งซึ่งใช้กับระบบ file ตามลำดับชั้นอยู่หลายคำสั่งคือ

MKDIR (Make directories) ใช้ในการสร้าง directory ย่อยขึ้นใหม่

CHDIR (Change directories) ใช้ในการเปลี่ยน directory ที่จะใช้เป็น working directory ต่อไป

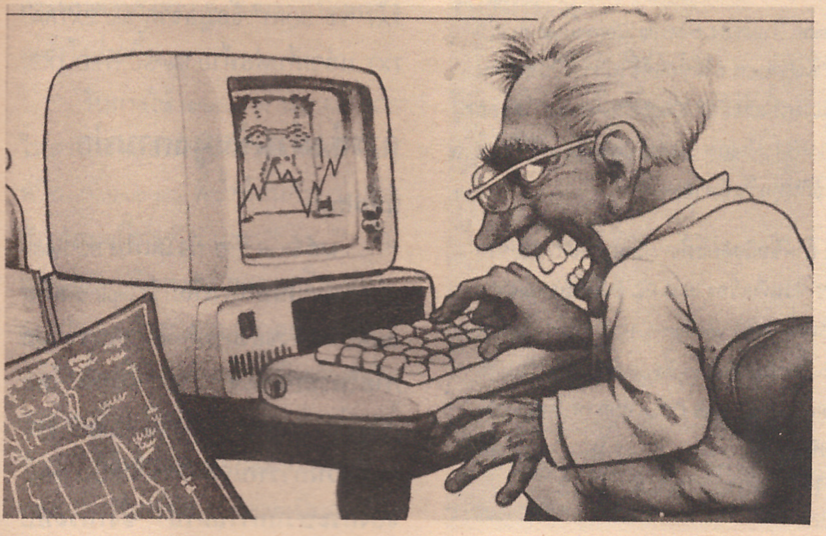
RMDIR (Remove directories) ใช้ในการลบ directory ทิ้ง

TREE ใช้แสดงโครงสร้างของ directories ในหลาย ๆ ระดับ

DIR ใช้แสดง directories ในระดับเดียว

ระบบ directory หลายระดับนี้อาจก่อให้เกิดปัญหาขึ้น เช่น ตามปกติโปรแกรมต่าง ๆ จะถูกจัดเข้าเป็นหมวดหมู่ตามลักษณะของโปรแกรมนั้น ๆ โดยโปรแกรม Compiler อาจจะรวมอยู่ใน directory ย่อยเดียวกัน ในขณะที่ Appli-cation program จะรวมอยู่ในอีกหนึ่ง directory ย่อย และโปรแกรม Linker อยู่ใน directory ย่อยที่สาม ในการใช้งานโปรแกรมทั้งสามนี้พร้อมกัน (ในการ Compile และ link โปรแกรม) จึงต้องเปลี่ยน default directory ไปมา ซึ่งไม่สะดวกอย่างยิ่ง MS-DOS ใช้คำสั่ง PATH เพื่อแก้ปัญหานี้ โดยพารามิเตอร์ของคำสั่ง Path จะเป็นรายชื่อของ directory ต่าง ๆ ที่ MS-DOS จะไปค้นหาเมื่อไม่พบโปรแกรมใด ๆ ใน directory ที่ใช้งานอยู่ในขณะนั้น ตัวอย่างต่อไปนี้จะแสดงถึงวิธีการแก้ปัญหาในการ Compile และ link โปรแกรมซึ่งจะต้องอ้างถึง file ที่อยู่ใน directory อื่น

PATH A:\SYSTEM,B:\COMPILER\PASCAL,B:\LINKER



คำสั่งนี้จะบอกให้ MS-DOS ทราบว่าเมื่อค้นหาโปรแกรมใด ๆ ไม่พบใน directory ที่ใช้งานอยู่ในขณะนั้น จะไปค้นหา directory ชื่อ SYSTEM ที่ drive A, directory ชื่อ COMPILER\PASCAL ที่ drive B และ directory ชื่อ LINKER ที่ drive B สังเกตว่าเราสามารถใส่ชื่อ drive ได้ในคำสั่ง PATH ซึ่งจะทำให้การค้นหา file กระทำได้กว้างขวางยิ่งขึ้น

การเพิ่มพูนประสิทธิภาพใน Input และ Output

การปรับปรุงระบบโครงสร้างของ file ใน MS-DOS 2.0 ที่สำคัญอีกอย่างหนึ่งก็คือ ความสามารถในการเปลี่ยนทิศทางของอุปกรณ์ Input, Output (I/O Redirection) MS-DOS 2.0 ได้นำความสามารถอันนี้มาจาก Xenix โดยมีลักษณะการใช้งานเหมือนกัน ประโยชน์ของการเปลี่ยนทิศทางของ I/O ก็คือ เราสามารถเปลี่ยน Input หรือ Output ซึ่งเดิมเป็น device มาตรฐาน เช่น Keyboard, จอ CRT หรือ Communication Port ให้เป็นได้ทั้ง file หรือ device ตัวอื่น ๆ การทำ redirection ที่ใช้กันบ่อยที่สุดอย่างหนึ่งก็คือ การเปลี่ยน Output จากจอภาพไปออกทาง Printer

ความสามารถในการเปลี่ยนทิศทางของ I/O นี้มีความสัมพันธ์อย่างใกล้ชิดกับแนวความคิดของ device driver (ที่ได้กล่าวไปแล้วในตอนต้น) นั่นคือ เมื่อเราใช้ device ตัวใหม่และติดตั้ง driver แล้ว เราสามารถเปลี่ยน I/O ที่มีอยู่เดิมไปยัง device ตัวใหม่นี้ได้โดยการใช้ recirection ดังกล่าว

การใช้ I/O redirection ในคำสั่งของ MS-DOS

การใช้ I/O redirection ใน Command line ทำได้โดยการใส่สัญลักษณ์ 3 ตัว

ความเป็นมาของ MS-DOS

MS-DOS เป็นโปรแกรมควบคุมระบบ (operating system) ที่พัฒนาขึ้นโดยบริษัท Seattle computer Products สำหรับเครื่องคอมพิวเตอร์ของบริษัทเองที่ใช้ 8086 เป็น CPU ชื่อเดิมของมันคือ 86-DOS และก็เป็นตัวเดียวกับ Z-DOS ที่ใช้กับเครื่อง Zenith นั่นเอง ความเป็นมาของ MS-DOS เกี่ยวพันอย่างมากกับการพัฒนา Software สำหรับคอมพิวเตอร์ที่ใช้ 8086 เป็น CPU

พฤษภาคม ค.ศ.1975 Seattle Computer ได้สร้างต้นแบบ CPU card ที่ใช้ 8086 สำหรับเสียบกับ S-100 BUS ขึ้นโดยหวังจะให้ใช้กับ CP/M-86 ที่ Digital Research กำลังพัฒนาอยู่และคาดว่าจะออกมาไม่เกินสิ้นปี 1979 ขณะเดียวกัน Microsoft ก็กำลังเร่งพัฒนาซอฟท์-แวร์สำหรับ 8086 อยู่เช่นกันโดยเริ่มจาก Disk BASIC (ซึ่งพัฒนาจาก interpreter ของ Microsoft หรือ MBASIC ที่เรารู้จักกันดีนั่นเอง) ที่มี Operating System อยู่ในตัว Microsoft ได้เลือกใช้ 8086 card ของ Seattle Computer Product ในการพัฒนาซอฟท์แวร์ของตนและได้นำออกแสดงใน National Computer Conferrence ในเดือนมิถุนายน 1979 นั่นเอง ซึ่งก็เป็นการเปิดตัวครั้งแรกของ Disk BASIC ที่ RUN ด้วย 8086

พฤศจิกายน ปีเดียวกัน Seattle Computer ก็ได้เริ่มผลิต 8086 ออกจำหน่ายอย่างจริงจังโดยมี Disk BASIC เป็นซอฟท์แวร์เพียงตัวเดียวที่ใช้กับมันได้ในขณะนั้น เวลาผ่านไปเรื่อย ๆ จนถึง เมษายน ค.ศ.1980 CP/M-86 ก็ยังไม่ออกมา Seattle Computer จึงตัดสินใจที่จะสร้าง DOS (Disk Operating System) ขึ้นใช้เองแทน เนื่องจากความล่าช้าของ CP/M-86 และความต้องการ Operating System ที่ใช้งานทั่วไปสำหรับ 8086 (ไม่ใช่ใช้เฉพาะกับภาษา BASIC อย่างใน Disk BASIC) ได้เพิ่มขึ้นอย่างรวดเร็ว

DOS Version แรกที่ Seattle Computer สร้างขึ้น มีชื่อเล่นว่า QDOS 0.10 เป็นรูปร่างขึ้นใน สิงหาคม ค.ศ.1980 QDOS ย่อมาจาก Quick and Dirty Operating System (อย่าสับสนกับ QDOS ของ Sinclair QL เพราะเป็นคนละตัวกัน) ทั้งนี้เพราะมันเป็นโปรแกรมที่เขียนขึ้นอย่างลวก ๆ และรีบร้อน (ใช้แรงงานโปรแกรมเมอร์ 2 คน-เดือน) แต่กลับทำงานได้ดีอย่างน่าประหลาด QDOS 0.10 นี้มีครบทุกอย่างที่จำเป็นสำหรับการพัฒนาโปรแกรมภาษา Assem blay นอกจากeditor เท่านั้น แล้วอีก 1 สัปดาห์ต่อมา editor ก็ถูกเติมเข้าไป โปรแกรม editor ตัวนี้มีชื่อว่า EDLIN หมายถึง Editor of LINE เพราะมันเป็นline editor (Edit โปรแกรมได้ทีละบรรทัด) แบบง่าย ๆ ซึ่งคาดกันว่าคงจะใช้ได้สัก 5-6 เดือนเท่านั้น (แต่ความจริงปรากฏว่ามันอยู่ได้นานกว่านั้นมาก โดยกลายเป็นส่วนหนึ่งของ MS-DOS ไป)

ในปลายปี ค.ศ. 1980 DOS version ใหม่ของ Seattle Computer ก็เสร็จคราวนี้เขาเรียกมันว่า 86 - DOS version 3.3 และ Microsoft ก็ได้ซื้อสิทธิในการเป็นตัวแทนจำหน่ายแต่เพียงผู้เดียวของ DOS ดังกล่าว และในเวลาไล่เรี่ยกับที่ Digital Research ได้ออก version แรกของ CP/M-86 ในเดือน เมษายน 1981 นั่นเอง Seattle Computer ก็ได้ออก 86-DOS version 1.0 ซึ่งคล้ายกับ MS-DOS ที่ใช้กันทุกวันนี้มาก

กรกฎาคม 1981 นั่นเอง Microsoft ก็ได้ซื้อลิขสิทธิ์ทั้งหมดของ 86-DOS จาก Seattle Computer และเปลี่ยนชื่อมันเป็น MS-DOS ส่วนที่เครื่อง IBM PC ซึ่งออกมาหลังจากนั้นไม่นาน ได้นำไปใช้นั้น ที่จริงแล้วก็นับได้ว่าเป็น 86-DOS version 1.14 นั่นเอง และทาง Microsoft ก็ได้ปรับปรุงต่อไปจนถึง version 1.25 ซึ่งเป็น version ที่ออกจำหน่ายพร้อมกับ IBM PC ในเดือนเมษายน 1982 ส่วน version ล่าสุดคือ MS-DOS ที่มีการเปลี่ยนแปลงไปจากเดิมอย่างมาก (เห็นได้จากการที่เปลี่ยนเลขที่ versionจาก 1 กว่า ๆ ไป 2 เลย) นั้นออกมาพร้อมกับเครื่อง IBM PC รุ่นใหม่คือ XT ในเดือนกุมภาพันธ์ 1983

ต่อไปนี้เข้าไปข้างหน้าชื่อ file คือ “ < ” “ > ” และ “ >> ” สัญลักษณ์ “ < ” จะบ่งบอกว่าให้ใช้ file นั้นเป็นแหล่ง Input มาตรฐาน สัญลักษณ์ “ > ” จะหมายถึงให้ใช้ file นั้นเป็น Output มาตร–ฐานโดยการสร้าง file นั้นขึ้นใหม่เพื่อเก็บ Output (หรือในกรณีที่ file นั้นถูกสร้างขึ้นมาก่อนแล้วก็ให้ใส่ Output ที่ได้ทับลงใน file นั้นเลย) ส่วน “ >> ” ก็หมายถึงให้ใช้ file นั้นเป็น Output มาตร–ฐานเช่นกัน แต่ให้ใส่ข้อมูล Output ที่ได้ต่อท้าย file โดยไม่ลบข้อมูลเดิมใน file

ตัวอย่างต่อไปนี้จะแสดงถึงวิธีใช้ I/O redirection ชนิดต่าง ๆ สังเกตว่า MS-DOS จะมอง device ต่าง ๆ เป็นเสมือน file แบบหนึ่ง เช่น printer จะเรียกเป็น file ชื่อว่า PRN

1. DIR> PRN คือการพิมพ์ directory ออกทางเครื่องพิมพ์แทนจอภาพ

2. DIR> DIRLIST.DAT เป็นการส่งรายชื่อ file ใน directory ไปเก็บไว้ใน file ชื่อ DIRLIST.DAT

3. SORT< UNSORT.TXT> UNSORT TXT จะ SORT ข้อมูลใน file IMSPRT.TXT เก็บลงใน file SORTED.TXT

4. SORT< EXTRA.TXT >> SORTED. TXT จะ SORT ข้อมูลจาก file EXTRA. TXT แล้วเก็บต่อท้าย file SORTED.TXT

การส่งผ่านข้อมูลตามท่อ (piping)

MS-DOS 2.0 ยังได้เพิ่ม I/O Redirection อีกแบบหนึ่งเรียกว่า pipe ซึ่งเป็นลักษณะหนึ่งของการติดต่อสื่อสารข้อมูลภายในระบบของ Xenix เช่นกัน โดย piping สามารถช่วยให้เรานำ Output จากกระบวนการหนึ่งส่งผ่านไปเป็น Input ของกระบวนการถัดไป ซึ่งในระบบ

UNIX แล้วทุก ๆ คำสั่งในระบบจะทำงานแบบขนาน (parallel processing) แต่ใน MS-DOS จะทำแบบต่อเนื่อง (sequentially) ตัวอย่างเช่น สมมุติว่าเราต้องการ Sort รายชื่อ file ใน directory ที่มี file type เป็น .ASM ก็สามารถเขียนคำสั่งเพียงบรรทัดเดียวดังต่อไปนี้

DIR | FIND ".ASM" | SORT> LIST ASM

คำสั่งข้างบนนี้มีความหมายเดียวกันกับชุดของคำสั่งต่อไปนี้

DIR> PIPE 0.$$$

FIND ".ASM"< PIPE 0.$$$> PIPE 1.$$$

ERASE PIPE 0.SSS

SORT< PIPE 1.$$$> LIST.ASM

ERASE PIPE 1.$$$

เครื่องหมาย "|" จะทำการเชื่อม Output file มาตรฐานของคำสั่ง ทางด้านซ้ายของมันเข้ากับ Input file มาตรฐานของคำสั่งทางด้านขวา ใน MS-DOS, file ที่ทำการเชื่อมจะถูกสร้างขึ้นจริง ๆ บน disk แต่ในระบบ UNIX แล้ว คำสั่งทั้ง 2 จะติดต่อกันโดยตรงโดยไม่ต้องสร้าง file ขึ้นบน disk แต่ถึงแม้กระนั้นการใช้ pipe ใน MS-DOS ก็จะช่วยลดการพิมพ์คำสั่งยาว ๆ ลงและทำให้คำสั่งดูเข้าใจง่ายขึ้น

ในการใช้ redirection นั้น MS-DOS ไม่สามารถยอมให้มีการกำหนด Input หรือ Output file ที่ซ้ำซ้อนกัน นั่นคือ สัญลักษณ์ "< " และ " >> " ไม่สามารถอยู่ติดทางซ้ายของเครื่องหมาย pipe (|) เพราะจะกลายเป็นการส่ง Output ออกไป 2 ทาง คือส่งไปเก็บทั้งใน file และส่งไปให้คำสั่งถัดไป สัญลักษณ์ " < " ก็ไม่สามารถอยู่ติดทางด้านขวาของ " | " เพราะจะกลายเป็นการรับ Input มาจาก 2 ทางคือ ทั้งจาก Output ของคำสั่งที่แล้วกับข้อมูลจาก file

ตัวกรอง (filter)

โปรแกรมที่เขียนขึ้นเพื่อใช้กับ pipe เรียกว่า "ตัวกรอง" (filter) โปรแกรมที่เป็นตัวกรองจะใช้ input มาตรฐานมาทำการสร้างข้อมูลขึ้นเพื่อส่งออกไปยัง Output มาตรฐาน MS-DOS ได้ผลิตตัวกรองขึ้น 3 โปรแกรมคือ FIND, SORT และ MORE FIND จะรับ parameter ซึ่งเป็นข้อมูลที่จะค้นหาใน Doubel quotes (" ") ค้นหาบรรทัดที่มีข้อมูลนั้นจาก Input มาตรฐาน แล้วส่งออกไปยัง Output มาตรฐาน MORE จะอ่านข้อมูลจาก Input มาตรฐานมาแล้วพิมพ์ออกทางจอภาพจนเต็มจอ แล้วรอรับการกด Keyboard เพื่อพิมพ์ข้อมูลหน้าต่อไป

การเปลี่ยนทิศทางของ I/O และตัวกรองเป็นเครื่องมือที่ programmer ทั่วไปสามารถนำไปประยุกต์ใช้งานกับ Application program ทั่ว ๆ ไป โดยเฉพาะอย่างยิ่ง เมื่อนำไปใช้กับการสร้าง batch file ซึ่งจะช่วยเพิ่มประสิทธิภาพขึ้นอย่างมาก

โปรแกรม Support ที่สำคัญบางโปรแกรม

CONFIG.SYS file นี้จะถูกอ่านโดย MS-DOS เมื่อเริ่มต้นระบบครั้งแรก ต่อไปนี้เป็นตัวอย่างของ Configuration file

BREAK = OFF

BUFFERS = 2

FILGS = 8

DEVICE = ANSI.SYS

SHELL = A: COMMAND.COM

BREAK option จะบอก MS-DOS ว่าจะตรวจสอบ CONTROL-C (Break) เฉพาะเมื่อตอนเรียกใช้ Console I/O เท่านั้น (โดยใช้ BREAK = OFF) หรือจะตรวจสอบ CONTROL-C ในทุกครั้งที่เรียกใช้ System call ของ MS-DOS (BREAK = ON) โดยชนิดหลังนี้จะมีประโยชน์ในการหยุดโปรแกรมบางจำพวก เช่น Compiler ซึ่งเรียกใช้ Console I/O น้อยมาก

BUFFERS option จะกำหนดจำนวนของ disk buffer ขนาด 512 byte ที่จะใช้ในระบบ

ข้อมูลที่ส่งเข้าออกใน disk จะถูกส่งผ่าน buffer เหล่านี้ MS-DOS จะฉลาดพอที่จะไม่ access disk ถ้าข้อมูลที่ต้องการอยู่ที่ buffer ใด buffer หนึ่งในจำนวนที่มีอยู่ทั้งหมด เราสามารถเพิ่มความเร็วในการทำงานขึ้นได้ เมื่อเพิ่มจำนวน Buffer ให้มากขึ้น โดยเฉพาะอย่างยิ่งเมื่อใช้โปรแกรมพวก Database

FILES option จะกำหนดจำนวน file ที่มากที่สุดที่สามารถเปิดได้ในเวลาเดียวกัน

DEVICE option ใช้ในการ load device driver เข้าไปรวมใน Operating System ซึ่งจะทำให้ผู้ใช้สามารถเรียกใช้ device ตัวใหม่ได้ option นี้มีประโยชน์อย่างมากในการดัดแปลง O.S. เพราะผู้ผลิตอุปกรณ์ทาง Hardware เพียงแต่เขียน device driver ที่ใช้กับอุปกรณ์ของตนขึ้นมาเท่านั้น ไม่จำเป็นต้องไปแก้ไข O.S. โดยตรง

SHELL option จะกำหนดให้ file COMMAND.COM เป็น default shell ของระบบ option นี้มีข้อดีที่สำคัญคือ เราสามารถใช้โปรแกรมอื่น ๆ เป็น default shell ได้ จึงเป็นไปได้ที่เราจะสร้างระบบ Menu-driven ขึ้นเป็น turn-key system โดยผู้ใช้อาจจะไม่ต้องติดต่อกับ MS-DOS เลย เพียงแต่เลือกการทำงานต่าง ๆ ที่ต้องการจากเมนูเท่านั้น

BACKUP และ RESTORE

แม้ว่าระบบ Hard disk ซึ่งมีระบบ Back up โดยใช้ tape หรือ catridge disk จะสามารถใช้ได้ใน MS-DOS ก็ตามแต่ floppy disk ก็ยังเป็นระบบการ Back up ที่ใช้กันเป็นส่วนใหญ่ MS-DOS 2.0 ได้สร้างโปรแกรมขึ้นเพื่อใช้กำหนดมาตรฐานในกระบวนการ Back up คือ BACK UP และ RESTORE ซึ่งมี Option ที่มีประสิทธิภาพอยู่หลายตัว ยกตัวอย่างเช่น

BACKUP C: WORKFILE ~SOURCE.* B: /S/M/A/D:18- 05-84

เป็นคำสั่งให้ใช้ floppy disk drive B. สำหรับการ backup ทุก ๆ extension ของ file ที่มีชื่อว่า SOURCE ใน drive C ใน directory ย่อย ชื่อ workfile

/S หมายถึงให้ Backup directory ย่อยทุกตัวของ SOURCE.* ด้วย

/M หมายถึงให้ Backup เฉพาะ file ที่ได้มีการ modified ไปแล้วเท่านั้น

/D: 18-05-84 หมายถึงให้ Backup เฉพาะ file ที่มีวันที่ ที่ update ครั้งหลังสุดเท่ากับ หรือหลังวันที่ 18 พฤษภาคม 1984

/A หมายถึงให้เพิ่ม file ที่จะ Backup เข้าไปใน drive B (ซึ่งอาจจะมีข้อมูลต่าง ๆ เก็บอยู่แล้ว)

คำสั่ง RESTORE เป็นการนำ Backup file กลับมาใช้งานใหม่ Option ต่าง ๆ ก็มีลักษณะคล้าย ๆ กับคำสั่ง BACKUP

RECOVER เป็นคำสั่งที่ช่วยในการนำ file กลับคืนมาจากการที่ directory ถูกทำลายไป หรือเสียหายบางส่วน คำสั่งนี้จะอ่าน directory และ พยายามที่จะสร้าง file ขึ้นมาใหม่ file ที่นำกลับคืนมาได้โดยวิธีนี้อาจมีขยะปะปนอยู่บ้าง จึงเป็นหน้าที่ของผู้ใช้ที่จะกำจัดข้อมูลที่ผิดพลาดนี้เอง โดยอาจจะใช้ WORD PROCESSOR เข้าช่วย

ASSIGN เป็นคำสั่งที่จะติดต่อ logical drive name เข้ากับ physical drive name มีประโยชน์ในการแก้ไขปัญหาของบางโปรแกรมที่ต้องการใช้ drive ซึ่งมีชื่อไม่ตรงกับที่เครื่องมี ยกตัวอย่าง เช่น โปรแกรมทำบัญชีซึ่งเป็นภาษา ASSEMBLY กำหนดให้ใส่แผ่นโปรแกรมใน drive A และแผ่นข้อมูลใน drive B แต่ถ้าต้องการใช้โปรแกรมนี้กับ Hard disk (ซึ่งตั้งชื่อเป็น drive E) แล้ว

คำสั่ง ASSIGN A = E, B = E

จะทำให้ Hard disk มองดูเหมือนกับเป็น drive A และ B และโปรแกรมนี้ก็สามารถ run ได้อย่างถูกต้องโดยไม่ต้องเปลี่ยนแปลงแก้ไขส่วนอื่น ๆ เลย

VERIFY คำสั่งนี้จะช่วยแก้ปัญหาที่เกิดกับระบบที่ใช้แผ่น disk ได้ดี การ set ให้ VERIFY ON จะสั่งให้ MS-DOS อ่านข้อมูลที่เพิ่งจะเขียนลงบน disk เพื่อให้แน่ใจว่าข้อมูลที่เขียนลงไปนั้นถูกต้อง

PRINT

คำสั่งนี้ช่วยให้เราสามารถทำ backgroud print spolling (ดูในกรอบ) ใน MS-DOS ได้อย่างมีประสิทธิภาพ โดยเราสามารถพิมพ์ข้อมูลออกทาง printer ในขณะเดียวกันก็ทำงานอื่น ๆ เช่น edit ข้อมูลไปด้วย คำสั่ง PRINT สามารถเก็บ FILE ที่จะพิมพ์ใน QUEUE ได้ถึง 10 files

ในตอนนี้ เราได้กล่าวถึงลักษณะโดยสังเขป MS-DOS 2.0 และ feature ใหม่ ๆ ที่เพิ่มเติมเข้ามา ซึ่งบางส่วนก็นำมาจาก Xenix (UNIX Version ของไมโครซอฟท์) สิ่งเหล่านี้ทำให้ MS-DOS เป็นโปรแกรมควบคุมระบบที่มีประสิทธิภาพสูง และใช้กันแพร่หลายมากที่สุดตัวหนึ่งในปัจจุบัน ในตอนต่อ ๆ ไปของบทความนี้ จะได้กล่าวถึงรายละเอียดต่าง ๆ ของ MS-DOS 2.0 เพิ่มเติมขึ้นอีก โปรดติดตามได้ในฉบับหน้านะครับ ■

Background Print Spooling

ตามปกติเมื่อ CPU ส่งข้อความไปยัง printer แล้ว line หนึ่ง ก็จะคอยจนกว่า printer จะพิมพ์จบ 1 line จึงค่อยส่ง line ต่อไป ในระหว่างที่ printer ยังพิมพ์ไม่เสร็จ CPU จะต้องรออยู่ในสภาพ-WAIT คือ execute คำสั่งไม่ได้ แต่ printer เป็น machanical device ที่มีความเร็วต่ำมากเมื่อเทียบกับอัตราการส่งข้อมูลของ CPU ดังนั้นถ้าข้อความยาว ๆ CPU จะต้องเสียเวลาคอยโดยเปล่าประโยชน์ วิธีแก้ไขในเรื่องนี้ที่ใช้กันอยู่มี 2 ทางคือ

- ใช้หน่วยความจำสร้างเป็น Buffer เพื่อรับข้อมูลทั้งหมดจาก CPU มาเก็บไว้แล้วค่อย ๆ ส่งให้ printer พิมพ์ไปเรื่อย ๆ วิธีนี้เสียค่าใช้จ่ายสูงเนื่องจากต้องติดตั้งอุปกรณ์เพิ่มเติม
- ใช้วิธี Background Print Spooling คือ CPU จะทำงานอื่น ๆ แล้วแต่ผู้ใช้จะต้องการไปเรื่อย ๆ ในขณะที่เจียดเวลาบางส่วนเท่าที่จะหาได้ (เช่นระหว่างช่วงเวลาที่คุณกดคีย์แต่ละครั้ง) ไปส่งข้อความที่จะพิมพ์ไปยัง printer ไปเรื่อย ๆ วิธีนี้จะได้ผลลัพธ์ช้ากว่าบ้าง แต่ไม่ต้องเสียค่าใช้จ่ายเพิ่มขึ้น ระบบ spooling จะทำงานอยู่เป็นเสมือนฉากหลัง (background) คุณจะสามารถทำงานต่าง ๆ ของคุณไปได้ตามปกติโดยไม่รู้สึกว่า CPU เจียดเวลาไปทำการพิมพ์แต่อย่างใด

โดย วสวี

SINCLAIR QL
ก้าวใหม่ของซินแคลร์

"ไมโครคอมพิวเตอร์ 16/32 บิทที่ถูกที่สุด"

คติพจน์ของเซอร์คลิฟ ซินแคลร์ มีอยู่ว่า ทำของถูกเข้าไว้ก็ขายได้เอง ซึ่งก็ได้พิสูจน์ให้เห็นแล้วจาก ZX 81 คอมพิวเตอร์ที่ราคาถูกและแพร่หลายที่สุดที่เรารู้จักกันดี ZX 81 เป็นคอมพิวเตอร์ขนาด 8 บิทที่ราคาต่ำกว่า 100 เหรียญ และตามมาด้วย ZX SPECTRUM คอมพิวเตอร์ขนาด 8 บิทอีกเช่นกันที่แสดงผลเป็นแบบสีได้หลายๆ สี แต่ในขณะนี้ซินแคลร์มีคอมพิวเตอร์รุ่นใหม่ที่น่าสนใจ ในราคาที่ถูกอย่างไม่น่าเชื่อออกมาอีกรุ่นหนึ่ง นั่นคือ Sinclair QL

QL ย่อมาจาก Quantum Leap อันหมายถึงก้าวกระโดดครั้งใหญ่ไปในอนาคตของซินแคลร์และเป็นก้าวแรกของบริษัทในตลาดคอมพิวเตอร์สำหรับธุรกิจขนาดเล็กด้วย QL ได้เริ่มออกจำหน่ายแล้วในอังกฤษ และในไม่ช้าก็คงมีจำหน่ายในอเมริกา (คราวนี้ไม่ใช่ภายใต้ยี่ห้อของ TIMEX อย่างเคย แต่บริษัทซินแคลร์วางแผนจะจัดการผลิตและจำหน่ายเอง) และก็คงจะเข้ามาในเมืองไทย เร็วๆ นี้



รูปพรรณของ QL

ขนาดภายนอกของ QL ก็ไม่ใหญ่โตอะไร ตามที่เห็นในรูปนี่แหละครับ กว้าง 5⅜ นิ้ว ยาว 18¾ นิ้ว และหนา 1¾ นิ้ว อุปกรณ์ทั้งหมดประกอบอยู่ในตัวถังขนาดดังกล่าวนี้เพียงชิ้นเดียว

CPU ของ QL ใช้ไมโครโปรเซสเซอร์ 68008 ของบริษัทโมโตโรล่า ซึ่งเป็นไมโครโปรเซสเซอร์ 16/32 บิท ตระกูลเดียวกับ 68000 ที่ใช้ใน Macin-

tosh ของ Apple หรือ XT/370 ของ IBM (อ่านเรื่องของ 68000 ในฉบับที่ 9) ทำงานที่ความถี่ 7.5 MHZ และยังมีไมโครโปรเซสเซอร์ 8049 ของอินเทลเป็นตัวทำหน้าที่ประจำต่างๆ เช่นการรับข้อมูลจากคีย์บอร์ด (เช่นเดียวกับใน GX-10 ของ EPSON), การสร้างเสียง, จัดการรับส่งข้อมูลจาก port RS-232 และระบบนาฬิกาบอกเวลาภายใน

นอกจากไมโครโปรเซสเซอร์สองตัวที่กล่าวแล้ว ภายใน QL ยังมีไอซีที่สร้างขึ้นเป็นพิเศษ (Custom IC) อีกสี่ตัวทำหน้าที่ควบคุม memory, หน่วย display, ระบบ Local Area Network ที่ติดตั้งมาในตัว ตลอดจนการทำงานของระบบเก็บข้อมูลโดยใช้เทป (Sinclaire Microdrive) อีก 2 ตัวที่สร้างมาภายใน QL แล้ว

เมื่อพูดถึง Microdrive ของ Sinclair อาจมีหลายท่านยังไม่รู้จัก ที่จริงแล้วมันก็คือเครื่องบันทึกข้อมูลที่ใช้เทปคาสเซทท์ขนาดเล็กพิเศษ (Microcassette) ที่ Sinclair ผลิตขึ้นเอง (ดูรูป) Microdrive นี้มีใช้กับเครื่อง SPCETRUM มาก่อนแล้ว Microcassette ดังกล่าว 1 ตลับ บรรจุเทปคุณภาพสูงยาว 200 นิ้ว เลื่อนผ่านหัวเทปด้วยความเร็ว 30 นิ้ว/วินาที ดังนั้นเวลาที่ใช้ในการค้นหาข้อมูลตลอดความยาวม้วนก็ประมาณ 7 วินาที ส่วนเวลาที่ใช้โดยเฉลี่ยก็ตกประมาณ 3.5 วินาที Microcassette 1 ตลับจะเก็บข้อมูลได้ประมาณ 100 K Byte และสามารถต่อ Microdrive เพิ่มได้อีก 6 ตัว รวมเป็น 8 ตัว เพื่อเก็บข้อมูลได้ถึง 800 K Byte

ทางด้านหน่วยความจำของ QL จะประกอบด้วย RAM 128 K Byte ที่สามารถเพิ่มเติมได้ถึง 640 K โดยการใช้ RAM Pack เสียบเพิ่มภายนอกทำนองเดียวกับ ZX 81 และ ROM 32-K Byte ซึ่งบรรจุ interpreter ภาษา BASIC ที่เรียกว่า Sinclair Super BASIC กับโปรแกรมควบคุมระบบ QDOS หน่วยความจำจะถูกแบ่งออกเป็น sector แต่ละ sector มีขนาด 512 ไบท์ (0.5 K) รวม 256 sector (รวมเป็นหน่วยความจำ 256×0.5 = 128 K)

คีย์บอร์ดของ QL เป็นแบบมาตรฐาน คือ จัดอักษรแบบ QWERTY มีขนาดใหญ่เท่ากับคอมพิวเตอร์ที่ใช้งานธุรกิจทั่วไป มีทั้งหมด 65 คีย์ (ไม่มี numeric key) รวมทั้ง function key 5 คีย์ คีย์ควบคุม cursor 4 ทิศทาง ภายใต้คีย์มีแผ่นรองเพื่อป้องกันอันตรายจากการที่ของเหลวเช่นน้ำ หรือกาแฟจะหกรด แต่ละคีย์จะส่งเสียง "คลิค" เพื่อแสดงให้รู้ว่ากดลงถึงที่แล้ว

กราฟฟิคของ QL มี 2 mode คือ 512×256 จุด 4 สีคือ ดำ, ขาว, แดงและเขียว หรือ 256×256 จุดกับอีก 4 สีที่เพิ่มเข้ามาคือ น้ำเงิน, ม่วง (magenta), เขียวอมน้ำเงิน (cyan) และเหลือง หน่วยความจำ RAM ที่ใช้ในการแสดงกราฟฟิคนี้มีขนาด 32 K ส่วนตัวอักษรและตัวเลข (Alphanumeric) แสดงผลได้ 2 แบบแล้วแต่จอที่ใช้คือถ้าใช้เครื่องรับโทรทัศน์ก็จะแสดงได้แค่ 25 บรรทัด 40 ตัวอักษร/บรรทัด ถ้าใช้มอนิเตอร์ก็จะได้ถึง 80 ตัวอักษร/บรรทัด ด้านข้างของ QL มีที่ต่อทั้งสัญญาณ Composite video สำหรับ TV และสัญญาณ RGB สำหรับมอนิเตอร์สี

สำหรับการต่อกับอุปกรณ์เพอริเฟอรัลต่างๆ QL มี RS-232 C serial port ให้ถึง 2 port ซึ่งเลือก baudrate ได้ตั้งแต่ 75 ถึง 19,200 baud, มี port สำหรับ joystick อีก 2 อัน และ port สำหรับการติดต่อกับ Local Area Network หรือ Sinclair SPECTRUM ได้ในอัตราถึง 100 K baud

มีอะไรใน ROM

โปรแกรมควบคุมระบบ (Operating System) ของ QL เรียกว่า QDOS ซึ่งมีสิ่งที่น่าสนใจคือ QDOS สามารถจัดการทำงานหลายอย่างพร้อมๆ กัน (Multitasking ได้และสามารถสร้าง window ขึ้นซ้อนกันบนจอภาพได้ แบบเดียวกับ Lisa หรือ Macintosh) ถึง 20 จอ จอเหล่านี้จะถูกเรียกใช้ได้โดยคำสั่ง WINDOW ใน Sinclair super BASIC ดังนั้นโปรแกรมหลายโปรแกรมก็อาจทำงานพร้อมๆ กันไปได้เมื่อถูกเก็บอยู่คนละจอกัน

ใน ROM นอกจาก QDOS แล้วก็มี super BASIC ของซินแคลร์ ซึ่งนอกจากจะมีคำสั่งตามธรรมดาทั่วไปแล้วยังมีคำสั่งเพิ่มเติมเข้ามาอีกมากมาย เช่น BAUD (เลือก Baud rate), BEEP (ตามด้วย parameter ต่างๆ เพื่อสร้างเสียงในลักษณะต่างๆ กัน), EXEC (สั่งให้ทำงานตามโปรแกรมหลายๆ โปรแกรมต่อเนื่องกัน), PAN (เลื่อน window ที่ใช้อยู่ไปทางซ้ายหรือขวา) และอื่นๆ อีกมาก

ซอฟแวร์สำหรับ QL

พร้อม ๆ กับการเปิดตัวของ Sinclair QL ทางบริษัทยังได้แนะนำแพคเกจ 4 อันที่จะเป็นเครื่องช่วยในการบุกเบิกตลาดของ QL ด้วย แพคเกจทั้ง 4 ตัวนี้สามารถใช้ขีดความสามารถในการแสดงผลเป็นสีของ QL ให้เป็นประโยชน์

ได้อย่างดี และรูปแบบการเก็บข้อมูลก็ทำให้สามารถใช้ข้อมูลร่วมกันได้ในระหว่างแพคเกจทั้งสี่

ตัวแรกที่จะพูดถึงคือ QL Quill ซึ่งเป็นโปรแกรม word processor ที่มีฟังชันต่างๆ ที่ช่วยอำนวยความสะดวกในการทำงานเช่น insert, erase, copy, move, search, replace, tabs, header, footnote, justification (จัดกั้นหน้า-หลังให้ตรงกันและตัดคำ) และอื่นๆ อีกมาก โปรแกรมนี้ใช้ได้ทั้งบรรทัดที่มี 40/60/80 และ 132 ตัวอักษรและยังสามารถโปรแกรมฟังชันคีย์ให้เป็นคำสั่งตามต้องการได้ด้วย ลักษณะของ QL Quill อีกประการหนึ่งคือสามารถแสดงผลบนจอได้เช่นเดียวกับที่จะพิมพ์ลงบนกระดาษ เช่น ถ้าคุณใช้ตัวอักษรตัวหนา (boldface) คุณก็จะเห็นมันเป็นตัวหนาบนจอเช่นเดียวกับที่จะพิมพ์ออกทาง printer

แพคเกจถัดมาก็คือ QL Abacus ซึ่งเป็นโปรแกรมกระดาษทดอิเลคโทรนิคส์ (electronic spreadsheet) ที่มีขนาดสูงสุด 256 แถว×64 คอลัมน์ ผู้ใช้สามารถตั้งชื่อแถวทั้งแถว คอลัมน์ทั้งคอลัมน์หรือเฉพาะเซลใดเซลหนึ่ง เพื่อใช้อ้างถึงในสูตรคำนวณที่จะป้อนเข้าไปก็ได้ การคำนวณให้ค่าถูกต้องถึง 16 หลัก และ spreadsheet หลายๆ ชุดก็สามารถนำมารวมกันได้ด้วย แต่ลักษณะเด่นที่สุดของ QL Abacus เห็นจะได้แก่การแบ่งจอออกเป็นช่องๆ หลายช่อง (multiple window) นอกจากนี้ก็มีการเปลี่ยนแปลงขนาดของแต่ละเซล และการใช้สัญญาณ prompt แก่ผู้ใช้เพื่อรอรับ input เข้าไปใช้ในการคำนวณใน worksheet ได้ด้วย

สำหรับแพคเกจอันที่สามก็เป็นซอฟท์แวร์ที่ใช้กับระบบฐานข้อมูลคือ QL Archive ซึ่งมีคำสั่งที่สามารถใช้ในการจัดระบบและค้นหาข้อมูลที่ต้องการได้มากกว่า 80 คำสั่ง รวมทั้งการ search, sort, select เฉพาะ record ที่ต้องการ, การประมวลผลกับข้อความหรือชุดของตัวอักษร (string manipulation) และการคำนวณทางคณิตศาสตร์ คำสั่งใน QL Archive จัดอยู่ในโครงสร้างแบบปิรามิดคือมีหลายระดับลดหลั่นกันไป คำสั่งที่ถูกใช้บ่อยๆ จะถูกจัดไว้ในระดับแรกๆ ซึ่งเรียกใช้ได้ทันที



"QL มี microcassett drive 2 ตัวอยู่ภายในใช้กับ microcassette ลักษณะดังในภาพซึ่งมีความจุ 100 K ต่อหนึ่งตลับ"

ส่วนคำสั่งที่ไม่ค่อยได้ใช้บ่อยนักจะถูกจัดไว้ในระดับถัดๆ ไป ซึ่งจะเรียกใช้ได้โดยผ่านทาง menu อีกทีหนึ่ง

แพคเกจสุดท้ายที่จะพูดถึงก็คือ QL Easel ซึ่งเป็นแพคเกจทางด้านกราฟฟิคที่มีประสิทธิภาพสูง ทั้งข้อความ (text) และรูปภาพ (graphic) สามารถจะนำมารวมกันได้ สีต่างๆ ที่จะใช้สามารถเลือกได้จากที่มีอยู่ 17 สี กับสีพิเศษอีกสีหนึ่งที่ผู้ใช้สามารถกำหนดเองได้ QL Easel นี้สามารถจะนำมาใช้รวมกับ QL Quill ในการสร้างเอกสารและรูปภาพประกอบได้โดยสะดวก

อนาคตของ QL

นอกจากแพคเกจทั้ง 4 นี้แล้วขณะนี้ Sinclair QL ยังไม่มีซอฟท์แวร์อื่นใดมาสนับสนุนอีก แต่ทางบริษัทก็แถลงว่าบรรดาผู้ผลิตซอฟท์แวร์ในประเทศอังกฤษหลายแห่งกำลังเร่งพัฒนาซอฟท์แวร์สำหรับ QL อยู่อย่างขะมักเขม้นเพื่อให้ทันกับการบุกเบิกตลาดในอเมริกาของ QL บรรดาซอฟท์แวร์เหล่านั้นก็มีอาทิเช่น Pascal Compiler, Terminal Emulator (โปรแกรมที่ทำให้ไมโครคอมพิวเตอร์ทำงานเป็นเสมือนเทอร์มินัลที่จะใช้ต่อกับเครื่องเมนเฟรมได้), Assembler (ตัวแปลภาษา Assembly) ของไมโครโปรเซสเซอร์ 68000 เป็นต้น ส่วนอุปกรณ์เพิ่มเติมทางฮาร์ดแวร์ที่จะออกมาในอนาคตก็มีแผงวงจร RAM ขนาด 0.5 Megabyte (512 K), A/D converter (ตัวแปลงสัญญาณดิจิตอลอนาลอก), hard disk, Modem, parallel printer และ IEEE-488 bus interface (IEEE-488 เป็นระบบการอินเตอร์เฟสที่ใช้กับงานควบคุมเครื่องมือหรือ instrumentation control)

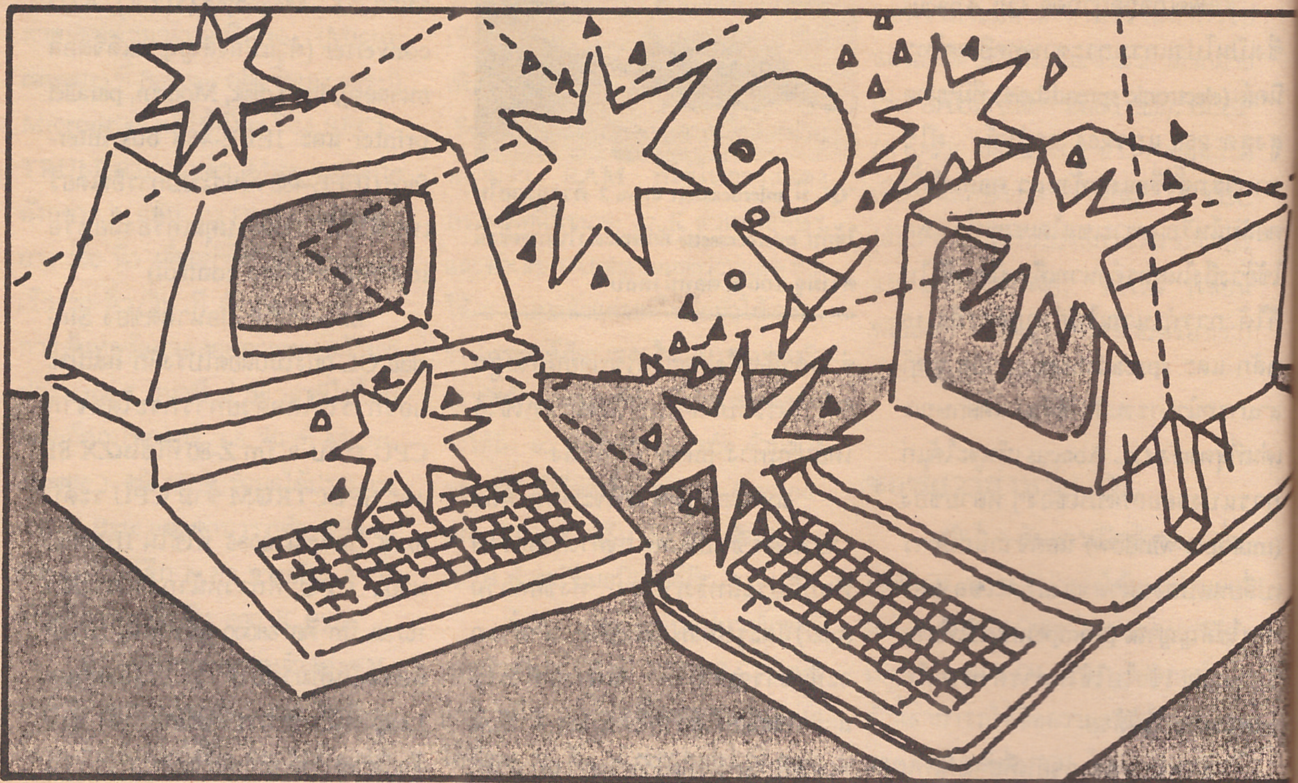
อย่างไรก็ตามอนาคตของ Sinclair QL ก็ยังไม่ปลอดโปร่งนัก ถึงแม้จะมองการณ์ไกลด้วยการกระโดดจาก CPU ระดับ 8 บิท Z-80 ที่ใช้ในZX 81 และ SPECTRUM ข้าม CPU ระดับ 8/16 บิท เช่น 8088 ที่ใช้ใน IBMPC มายัง CPU ที่ก้าวหน้ากว่าในระดับ 16/32 บิท คือ 68000 แล้วก็ตาม (บางทีนี่อาจเป็นที่มาของคำว่า Quantum Leap หรือการกระโดดไกลก็ได้) ทั้งนี้ด้วยสาเหตุที่สำคัญหลายประการ เช่น

● การใช้ microcassette ที่มีความจุเพียง 100 K เป็นอุปกรณ์ในการเก็บข้อมูลเพิ่มเติม ซึ่งต่างจากแนวโน้มของตลาดที่นิยมใช้ floppy disk ขนาด 5¼ นิ้วและ 8 นิ้ว อยู่เดิมซึ่งมีความจุ 300-600 K หรือแนวโน้มใหม่ที่จะนิยมใช้ดิสค์ขนาด 3½ นิ้วของ SONY ที่มีความจุ 400 K ถึงแม้ Microdrive ของ Sinclair จะมีประสิทธิภาพสูง แต่ก็ยังมีความเร็วในการถ่ายเทข้อมูลต่ำกว่า disk อยู่ดี ถึงแม้การใช้ Microdrive จะทำให้ราคาของ QL ต่ำ แต่ในการใช้งานทางธุรกิจนั้นย่อมคำนึงถึงประสิทธิภาพไม่น้อยไปกว่าราคา

● โปรแกรมควบคุมระบบของ QL ใช้ QDOS ซึ่งเป็นโปรแกรมใหม่ที่ต่างจากความนิยมในตลาดซึ่งส่วนใหญ่เป็น MS-DOS (สำหรับ IBM PC) ทำให้ยังไม่มีซอฟท์แวร์สนับสนุนในตลาดมากนัก

● ประการสำคัญก็คือภาพพจน์และชื่อเสียงของ Sinclair เอง ที่ติดอยู่กับลักษณะของ "ของเล่น" มากกว่า "ของจริง" เป็นเรื่องยากที่จะเปลี่ยนทัศนคติของผู้คนทั่วไปที่มีต่อ Sinclair จากผู้ผลิตคอมพิวเตอร์ระดับต่ำที่สุดสำหรับการศึกษาและการใช้งานในบ้านมาเป็นการใช้งานจริงในธุรกิจขนาดเล็ก นอกจากนี้ในกรณีของการใช้งานในบ้านเป็น home computer ก็อาจจะ "ดีเกินไป" เช่น ปกติแล้วถ้าเป็นการใช้ในบ้าน ผู้ซื้อมักจะใช้จอโทรทัศน์แทนที่จะใช้ RGB Monitor (เพราะราคาของ RGB Monitor หรือมอนิเตอร์สีนั้นสูงมากเมื่อเทียบกับราคมที่ถูกอย่างไม่น่าเชื่อของ QL ดังนั้นขีดความสามารถทางด้านกราฟฟิคของ QL จึงไม่ได้ถูกใช้อย่างเต็มที่เป็นต้น และตลาดในระดับ home computer ก็ไม่ใช่ว่าจะปลอดโปร่งเท่ากับในยุคของ ZX 81 เพราะขณะนี้บรรดายักษ์ใหญ่ทั้งหลายต่างกำลังแข่งขันกันอยู่ในระดับนี้ด้วย QL จะต้องพบกับการแข่งขันทั้งกับ IBM PC jr, Macintosh ของ Apple, Adam ของ Coleco และ Commodore ในระดับต่างๆ กัน

ไม่ว่าอนาคตของ QL จะเป็นอย่างไร การปรากฏตัวของ Sinclair QL ก็นับเป็นเหตุการณ์ที่ควรแก่การสนใจ ในฐานะของความพยายามที่จะนำเทคโนโลยี่ระดับ 16/32 บิทไปสู่ผู้ใช้ให้ได้กว้างขวางที่สุดในราคาที่ถูกที่สุดตามแบบฉบับของซินแคลร์ และอย่างน้อยก็พอจะคาดการณ์ได้ว่า Sinclair QL คงสามารถแทรกตัวอยู่ในตลาดได้ในระดับหนึ่งโดยมีส่วนแบ่งตลาดในอัตราพอสมควร ซึ่งจะมากน้อยเพียงไรก็ขึ้นอยู่กับกลยุทธทางการตลาดของทางบริษัทเองมากกว่าจะขึ้นอยู่กับเทคโนโลยี่ใช้ในผลิตภัณฑ์เสียอีก ■



SOFTWARE REVIEW

Electronic Spreadsheet

โดย ดนัยวัฒน์ กิตตินันท์

สวัสดีครับท่านผู้อ่าน ท่านคงจะรู้สึกแปลกตาแปลกใจกับคอลัมน์ SOFTWARE REVIEW ที่ท่านกำลังอ่านอยู่นี้คอลัมน์นี้เป็นคอลัมน์ใหม่ซึ่งจะเสนอบทความเกี่ยวกับ-SOFTWARE ต่าง ๆ ที่น่าสนใจ และคิดว่าเป็นประโยชน์กับผู้ที่สนใจทางด้านไมโครคอมพิวเตอร์ รูปแบบของบทความที่นำมาเสนอให้คอลัมน์นี้จะเป็นเรื่องราวทั่ว ๆ ไปของSOFTWARE ตัวนั้น ๆ รวมทั้งคุณสมบัติและความสามารถที่ดีเด่นหรืออาจจะเป็นบทความที่เป็นการเปรียบ-เทียบความสามารถข้อดี ข้อด้อย ระหว่าง SOFTWARE ประเภทเดียวกันเพื่อที่ผู้อ่านจะได้ใช้เป็นข้อมูลในการเลือกใช้ SOFTWARE ให้เหมาะสมกับงานที่ต้องการ ถ้าหากท่านผู้อ่านมีความคิดเห็นหรือข้อเสนอแนะอย่างไรก็ลองเสนอมาให้ทางผู้จัดทำรับฟังดูบ้างนะครับ

ในฉบับนี้ จะเป็นการแนะนำให้รู้จักแนะนำให้รู้จักและเปรียบเทียบระหว่าง SOFTWARE ประเภทหนึ่งซึ่งเป็นที่รู้จักกันเป็นอย่างดีในแวดวงไมโคร คือ SOFTWARE ที่เรียกกันว่า electronic spreadsheet หรือกระดาษทดอิเลคทรอนิคส์นั่นเอง ท่านที่คุ้นเคยกับซอฟท์แวร์ประเภทนี้ดีแล้วอาจจะข้ามไปดูตารางเลยก็ได้ แต่สำหรับท่านที่ยังไม่คุ้นเคย ก่อนอื่นเราก็จะมารู้จักเจ้ากระดาษทดอิเลคทรอนิคส์แผ่นนี้ให้ดีก่อนดีกว่านะครับ

ทำไมจึงเรียกว่า กระดาษทดอิเลคทรอนิคส์?

เห็นจะเป็นเพราะว่า SOFTWARE ประเภทนี้ มีลักษณะเหมือนกับกระดาษทดขนาดใหญ่ ซึ่งต่อไปจะขอเรียกว่า แผ่นงาน หรือ WORKSHEET และสำหรับผู้ใช้เสมือนกับว่ามีอุปกรณ์เครื่องเขียน คือ ปากกา ดินสอ และยางลบที่จะใช้ในการ เขียนลบ แก้ไข ข้อมูล ข้อความ สูตร (FORMULAR) ในการคำนวณลงบนกระดาษทดแผ่นนี้ แต่แผ่นงานหรือกระดาษทดนี้มีความสามารถเหนือกระดาษทดธรรมดาอีกมากมายมหาศาล เช่น การเปลี่ยนแปลงข้อมูลบางส่วนจะทำให้ข้อมูลอื่น ๆ บนแผ่นงานที่มีความสัมพันธ์เกี่ยวข้องเปลี่ยนแปลงตามไปโดยอัตโนมัติ ในขณะที่กระดาษทดธรรมดาเมื่อผู้ใช้แก้ไขข้อมูลหนึ่งก็ต้องตามไปแก้ข้อมูลที่เกี่ยวข้องอื่น ๆ เอง เขียนมาถึงแค่นี้ผู้อ่านคงจะพอมองเห็นภาพของ SOFTWARE ประเภทนี้กันบ้างแล้ว ต่อไปลองมาดูกันว่าแผ่นงานที่ว่านี้จริง ๆ แล้วมีรูปร่างลักษณะเป็นอย่างไร.

ลักษณะของแผ่นงาน

แผ่นงานนี้จะแบ่งออกเป็นส่วนย่อยที่เรียกว่า เซล เซลแต่ละเซลที่มีชื่อเรียกซึ่งอ้างอิงจากแถวและคอลัมน์ เช่น เซลที่อยู่ในคอลัมน์ A แถวที่ 1 มีชื่อว่า เซล A1 เซลที่อยู่ในคอลัมน์ C แถวที่ 2 เรียกว่า C2 ดังรูป

	A	B	C	D	E
1	A1				
2			C2		
3					
4					
5					

ขนาดของเซลและจำนวนเซล ซึ่งก็คือขนาดของแผ่นงานนั่นเอง ขึ้นอยู่กับชนิดของ SOFTWARE แต่ละตัว ใน

แต่ละเซลนี่แหละที่เราใช้ในการเก็บข้อความ ข้อมูล และสูตรในการคำนวณลงไป จะเห็นว่าเมื่อแผ่นงานมีจำนวนเซล, มาก ๆ คือมีจำนวนใหญ่ จะไม่สามารถแสดงภาพ แผ่นงานทั้งหมดบนจอได้ จึงแสดงได้เพียงบางส่วนของแผ่นงานเท่านั้น และผู้ใช้สามารถกำหนดให้จอภาพแสดงส่วนของแผ่นงานที่ต้องการบนจอได้ เมื่อรู้จัก spreadsheet รู้จักแผ่นงานแล้ว คราวนี้ก็จะลองนำแผ่นงานนี้มาใช้งานจริง ๆ กันดูบ้าง

ตัวอย่างการใช้งาน

เพื่อจะได้รู้จัก electronic spreadsheet ให้ดีขึ้น ลองมาดูตัวอย่างการใช้งานง่าย ๆ กัน สมมุติว่า ร้านขายรองเท้าที่ทันสมัยแห่งหนึ่ง (เพราะมีไมโครคอมพิวเตอร์ใช้) ต้องการทำบัญชีแสดงการขายรองเท้ายี่ห้อต่าง ๆ ในแต่ละเดือน มีข้อมูลที่ต้องแสดงคือ ยี่ห้อรองเท้า จำนวนที่ขายราคาขาย จำนวนเงินที่ขายได้ของรองเท้าแต่ละยี่ห้อ ยอดขายรวมทั้งหมด จำนวนเงินที่ขายได้ทั้งหมด เราก็นำมาเขียนลงบนแผ่นงานของเราดังรูป

	A	B	C	D	E
1	ITEM	QUANTITY	PRICE	AMOUNT	
2	LOTTO	10	950	B2 × C2	
3	NIKE	15	500	B3 × C3	
4	PUMA	18	800	B4 × C4	
5					
6					
7	TOTAL QUANTITY	B2 + B3 + B4	TOTAL AMOUNT	D2 + D3 + D4	

เซล A2,A3,A4 เป็นยี่ห้อต่าง ๆ ของรองเท้าที่ขาย

เซล B2,B3,B4 เป็นจำนวนที่ขาย

เซล C2,C3,C4 เป็นราคาของรองเท้าแต่ละยี่ห้อ

ในเซล D2 มีสูตร B2×C2 คือจำนวนรองเท้า LOTTO ที่ขายคูณกับราคาขาย ซึ่งก็คือ จำนวนเงินที่ได้จากการขายรองเท้า LOTTO นั่นเองในเซล D3,D4 ก็เช่นเดียวกับ D2

ในเซล B7 มีสูตร B2+B3+B4 คือ ยอดรวมของรองเท้าทุกยี่ห้อที่ได้ขายไป

เซล D7 มีสูตร D2+D3+D4 คือ จำนวนเงินรวมที่ได้จากการขายรองเท้าทุกยี่ห้อ

แผ่นงานนี้จะถูกเก็บไว้ในหน่วยความจำ และเมื่อนำมาแสดงเซลที่มีสูตรอยู่จะถูกแทนที่ด้วยตัวเลขที่ได้จากการคำนวณตามสูตรดังกล่าว ดังรูป ส่วนเซลที่เป็นตัวอักษรก็จะไม่มีการเปลี่ยนแปลงใด ๆ

	A	B	C	D	E
1	ITEM	QUANTITY	PRICE	AMOUNT	
2	LOTTO	10	950	9500	
3	MIKE	15	500	7500	
4	PUMA	18	800	14400	
5					
6					
7	TOTAL QUANTITY	43	TOTAL AMOUNT	31400	

เมื่อมีการแก้ไขข้อมูลเช่น ราคาขายของ NIKE เปลี่ยนเป็น 600 ก็จะทำให้ข้อมูลในเซลที่เกี่ยวข้องคือ D3 และ D7 เปลี่ยนแปลงตามไปโดยอัตโนมัติ ซึ่งก็คือจำนวนเงินที่ได้จากการขายรองเท้า NIKE และจำนวนเงินที่ขายได้เปลี่ยนไป

สำหรับวิธีการและคำสั่งในการเขียนข้อมูลลงในเซล การแก้ไข การเก็บแผ่นงานลงในหน่วยความจำ การคำนวณ และแสดงออกทางจอภาพ ขึ้นอยู่กับชนิดของ SOFTWARE ตัวนั้น ๆ ซึ่งหาดูได้จากคู่มือการใช้งานของ SOFTWARE ตัวนั้น ส่วนที่ได้อธิบายไปข้างต้นเป็นเพียงในหลักการในการใช้งานเท่านั้น

SOFTWARE ที่เราจะนำมาเปรียบเทียบกันในฉบับนี้เป็น SOFTWARE ประเภท electronic Spreadsheet 8 โปรแกรมด้วยกัน บางโปรแกรมเป็นที่รู้จักกันดี เช่น VISICACL, SUPERCALC หรือ MULTIPLAN แต่บางโปรแกรมอาจจะไม่ค่อยคุ้นหูนัก ทั้ง 8 โปรแกรมเป็น SOFTWARE สำหรับเครื่องไมโครคอมพิวเตอร์ 8 bit เท่านั้น แต่ละโปรแกรมมีรายละเอียดดังนี้

ชื่อ	บริษัทผู้ผลิต	ราคาจำหน่าย ใน U.S.A.
MAGICALC	ARTSCI	$ 149.95
MICROPLAN	CHANG LABS	$ 495.00
MULTIPLAN	MICROSOFT CORP.	$ 250.00
PERFECT CALC	PERFECT SOFTWARE INC.	$ 295.00
PRO CALC	SOFTWARE PRODUCT Intel., Inc.	$ 350.00
SUPERCALC	SORCIM	$ 295.00
TARGET: FINANCIAL MODELING	CORNSHARE TARGET SOFTWARE, Inc.	$ 325.00
VISICALC: ADVANCE VERSION	VISICORP	$ 400.00

คุณสมบัติความสามารถ ฟังก์ชันใช้งานต่าง ๆ ของแต่ละโปรแกรมดูได้จากตารางเปรียบเทียบ เครื่องหมาย ● ในตารางแสดงว่าโปรแกรมนั้นมีความสามารถดังกล่าว สำหรับบางฟังก์ชันขอทับศัพท์เป็นภาษาอังกฤษไว้เช่นใน FINANCIAL FUNCTION เพื่อไม่ให้ผิดไปจากความหมายเดิม

บทความนี้คงทำให้ผู้อ่านได้รู้จัก electronic spreadsheet ดีขึ้นและเป็นแนวทางในการนำ SOFTWARE ประเภทนี้มาใช้งาน หวังว่าโอกาสหน้าคงจะได้พบกันอีก สำหรับในฉบับนี้ขอสวัสดีกันแค่นี้ก่อนนะครับ. ■

อนาคตของ Spreadsheet

นับตั้งแต่ Electronic Spreadsheet ตัวแรกคือ VisiCalc โดย Visicorp ได้ออกจำหน่าย มันได้ทำยอดขายที่สูงลิ่ว และกระตุ้นอุตสาหกรรมซอฟท์แวร์ให้หันมาทุ่มเทความพยายามในการพัฒนา Spreadsheet กันอย่างมากมาย ในยุคแรกนั้น VisiCalc เป็นโปรแกรมที่ทำให้ Apple II ขายดีตามไปด้วย คนซื้อ Apple II เพราะอยากใช้ VisiCale มีไม่น้อย จุดอ่อนของ VisiCalc ก็คือ มันไม่สามารถใช้กับเครื่องที่ใช้ CP/M เป็นโปรแกรมควบคุมระบบได้

SUPERCALC เป็น Electronic Spreadsheet ที่ออกมาเพื่ออุดช่องว่างอันนี้ โดยบริษัท SORCIM SUPERCALC สามารถใช้ภายใต้ CP/M และยังมีความสามารถพิเศษเหนือกว่า VisiCalc ขึ้นไปอีก นับว่า SUPERCALC เป็น Spreadsheet ที่ประสบความสำเร็จมากอีกตัวหนึ่ง ต่อจากนั้นก็มี Spreadsheet อื่น ๆ ตามออกมาอีกมาก บางตัวก็เป็น general purpose คือใช้งานทั่ว ๆ ไป บางตัวก็มุ่งหมายให้ใช้งานทางด้านใดด้านหนึ่งโดยเฉพาะ เช่น มีการเพิ่มฟังก์ชันทางด้าน Financial เข้าไป

ยุคที่สองของ Spreadsheet เริ่มต้นเมื่อมี VisiCalc : Advance version อันเป็น VisiCalc ตัวใหม่ที่เพิ่มพูนความสามารถจากตัวเดิมขึ้นอีกออกมา และพบกับการแข่งขันจาก $SUPERCALC^2$ ที่ปรับปรุงขึ้นอีกเช่นกัน นอกจากนี้ยังมี Multiplan ของ Microsoft ซึ่งนับได้ว่าเป็น Spreadsheet ที่ประสบความสำเร็จสูงสุดตัวหนึ่ง Spreadsheet 3 ตัวนี้คือตัวที่เราได้เปรียบเทียบไว้ในตาราง *และอาจกล่าว ได้ว่าเป็น Electronic Spreadsheet ล้วน ๆ รุ่นสุดท้าย* เพราะถัดจากนี้แนวโน้มในการพัฒนาซอฟท์แวร์กำลังมุ่งไปสู่ integrated software คือซอฟท์แวร์ตัวเดียวทำได้หลายงาน ทั้ง Word processor, Data Base Management, Spreadsheet และอื่น ๆ หรือไม่ก็เป็น integrating environment คือโปรแกรมที่ควบคุมให้ซอฟท์แวร์หลาย ๆ ตัว (แต่ละตัวทำหน้าที่เฉพาะอย่าง) ทำงานร่วมกัน ส่งผ่านข้อมูลระหว่างกัน และแสดงผลทาง Window หลาย ๆ อันบนจอพร้อมกัน เช่น Visicorp ออก VisiOn ซึ่งเป็น integrating environment ส่วน SORCIM ก็ออก $SUPERCALC^3$ ซึ่งเป็น integrated software คือรวมเอาฟังก์ชันทางด้าน Business graphic ไว้ด้วย ส่วน Microsoft ก็ออก Microsoft window ซึ่งเป็น integrating environment สำหรับซอฟท์แวร์หลาย ๆ ตัวเช่นเดียวกับ VisiOn

SPREADSHEETS

	VISCALC ADVANCED VERSION	SUPERCALC²	MULTIPLAN
คอลัมน์/แถว	63/254	63/254	63/255
ขนาดหน่วยความจำต่ำสุดที่ต้องใช้	128K	64K	64K
ขนาดหน่วยความจำสูงสุดที่จะใช้ได้	ไม่จำกัด		64K(CP/M); 544K (non-CP/M)
เครื่องคอมพิวเตอร์ที่ใช้*	2, 3, 7	1, 2, 7, 9	1, 2, 3, 6, 7, 9
คุณสมบัติของเซล :			
การใช้สูตร (FORMULAR) ในเซล	●	●	●
การทำให้ข้อมูลในเซลไม่ปรากฏ	●	●	
ป้องกันข้อมูลในเซลจากการแก้ไขหรือเปลี่ยนแปลงโดยไม่ตั้งใจ	●	●	●
กำหนดให้มีเฉพาะข้อความในเซลใด ๆ	●		
กำหนดให้มีเฉพาะข้อมูลที่เห็นจำนวนเลขในเซลใด ๆ	●		
กำหนดค่า TAB ในเซล	●		
ALIGN ข้อมูลในเซล L = ซ้าย R = ขวา	L, R	L, R	L, R
SET GUTTER L = ซ้าย R = ขวา	L, R		
พิมพ์อักษรซ้ำในเซล			
Center text ใน Column	●		●
ALIGN TEXT ใน COLUMN (L = left, R = right)	L, R	L, R	L, R
เลขบวกนำหน้าด้วย + เลขลบนำหน้าด้วย −	●		
เลขลบนำหน้าด้วย − เลขบวกไม่มีเครื่องหมาย	●	●	●
แสดงเลขลบ ด้วยวงเล็บ ( )	●	●	
แสดงเลขลบด้วย CREDIT (CR) จำนวนบวกด้วย DEBIT (DR)	●		
แสดงเครื่องหมาย COMMA ในจำนวนเลขยาว ๆ	●		
แสดงทศนิยมเป็น ศูนย์	●		●
ไม่ต้องแสดงทศนิยม	●	●	
หาค่าเปอร์เซนเตจ	●	●	●
กำหนดจำนวนทศนิยมในการแสดงค่า	●	●	●
แสดงค่าเป็น INTEGER	●	●	●
แสดงค่าเป็น Scientific NOTATION	●	●	●
แสดงค่าใน GRAPHIC FORMAT nat	●	●	●
เก็บข้อความซึ่งมีความยาวกว่าขนาดคอลัมน์ของเซล			●
LOCK ONLY CELL WITH TEXT OR FORMULAR			●
กำหนดชื่อให้กับเซลเพื่อใช้ใน COMMAND หรือใน FORMULAR			●
แสดงค่าในรูปของเลขยกกำลัง		●	
เติมช่องว่างที่ทั้งสองข้างของข้อความด้วยตัวอักษร			
INTERFACE สำหรับการแลกเปลี่ยนข้อมูลกับ SPREADSHEET	Data Interchange Format (DIF)	Super Data Interchange (SDI)	Symbolic Link File Format (SYLK)
CONDITIONALS:			
IF-THEN	●	●	●
AND, OR, NOT	●	●	●
ISERROR และ ERROR	●	●	●
ISNA และ NA	●	●	●
TRUE, FALSE	●		●
CHOOSE	●		
LOOKUP	●	●	●
INDEXED TABLE			●
VECTOR LOOKUP			
ฟังก์ชั่นทางสถิติ	ไม่มี	ไม่มี	ส่วนเบี่ยงเบนมาตรฐาน (SD)
Graphics	only rudimentary bar graphs using *s	only rudimentary bar graphs using *s	only rudimentary bar graphs using *s
การสามารถในการ SORT	●	ทำแถวละคอลัมน์	แถวอย่างเดียว
Windows	สูงสุด 2	สูงสุด 2	สูงสุด 8

*1 = Apple II +; 2 = Apple IIe; 3 = Apple III; 4 = Chameleon; 5 = Columbia; 6 = Commodore C64; 7 = IBM PC; 8 = Kaypro; 9 = CP/M-80/85; 10 = CP/M-86; 11 = MP/M; 12 = UCSD Pascal

	MAGICALC	PERFECT CALC	PROCALC	TARGET: FINANCIAL MODELING	MICROPLAN
	63/254	52/255	32/192	999/5000	20/50
	48K	56K	56K	64K	64K
	ไม่จำกัด	64K	ขึ้นอยู่กับขนาดของ DISK	64 (CP/M) อื่น ๆ ไม่จำกัด	64K (CP/M)
	1, 2	2, 4, 5, 7, 9	7, 12	7, 10	1, 7, 10, 11
		●		●	●
	●		●		
	●	●	●		
	●				
				●	
	L, R	L, R	L, R		
				L	
				L, R	
		●	●		
	L, R	L, R	L, R	R (heads, foots only)	R
	●	●	●	●	
		●	●	●	
			● (C and D)		
		●	●	●	●
				●	●
					●
			●	●	
	2 places	●	●	●	● (0-3)
	●	●	●	●	●
		●			
	●	●	●		
				N/A	
				●	
			●	●	
			●		
	Data Interchange Format (DIF)	none	interfaces to Logi-Quest, DBMS	none	link module
	●	●	●	●	
	●	●	●		
	●				
	●				
	●				
	●				
	●	●	●		
			●		
	ไม่มี	ไม่มี	ไม่มี linear estemation	ไม่มี	ค่า DELTA, เปอร์เซนต์การเพิ่ม,- mean, Sigma, ความแปรปรวน moving average, smooth
	only rudimentary bar graphs using *s	only rudimentary bar graphs using *s	only rudimentary bar graphs using *s	none	none
	ไม่มี	ไม่มี	เฉพาะคอลัมน์	ไม่มี	ไม่มี
	สูงสุด 2	สูงสุด 2	สูงสุด 6	สูงสุด 4	ไม่มี

Spreadsheets

	VISCALC ADVANCED VERSION	SUPERCALC²	MULTIPLAN
FINANCIAL FUNCTIONS:			
Interest on discount rate	●		
Payment amount and Current value	●		
Future value	●		
Internal rate of return	●		
Net current value		●	●
Declining balance depreciation			
Sum of year's depreciation			
Annuity current value			
Straight line depreciation			
Double declining value depreciation			
Double declining balance with automatic switchover to straight line schedules			
Period by period depreciation			
Discount cash flows			
Loan schedule (annual, quarterly, monthly)			
Interest schedule on loan			
Select tax computation schedule			
Taxes using tax table and pretax earnings			
Savings balances and Percentage ratios			
Percent of total value			
Percentages based on rates			
ฟังก์ชั่น ทางคณิตศาสตร์ :			
ค่าสัมบูรณ์ (ABSOLUTE)	●	●	●
ค่าเฉลี่ย (AVERAGE)	●	●	●
ลอกการิธัมฐานสิบ	●	●	●
ลอกการิทึมฐานธรรมชาติ	●	●	●
Count	●	●	●
Cumulative total			
Ceiling			
ผลต่าง (DELTA)			●
ยกกำลัง (EXPONENT)	●	●	●
Floor			
Grow by			
Integer portion	●	●	●
Length			●
ค่าต่ำสุดและสูงสุด	●	●	●
Midpoint			●
Modulo	●	●	●
Negate			
Power			
Plug			
ปัดเศษ	●	●	●
หาเครื่องหมาย (SIGN)			●
รากที่สอง	●	●	●
ผลรวม (SUM)	●	●	●
Sum of products	●		
Total			
Weight			
บวกด้วยค่าคงที่			
ลบด้วยค่าคงที่			
หารด้วยค่าคงที่			
คูณด้วยค่าคงที่			
ARCCOS และ ARCSIM	●	●	
Atan	●	●	
Cos and Sin	●	●	●
Tan	●	●	●
พาย (Pi)	●	●	●
การพิมพ์รายงาน			
เว้นบรรทัด และเว้นเมื่อขึ้นหน้าใหม่	●	●	●
มี COLUMN/ROW TITLE ในแต่ละ Page	●		

MAGICALC	PERFECT CALC	PROCALC	TARGET: FINANCIAL MODELING	MICROPLAN
		●		
		●		
				●
●				
		●		
		●		●
		●		
				●
				●
				●
				●
				●
				●
				●
				●
				●
				●
				●
				●
●	●	●		●
●	●		●	
●	●			●
●	●			●
●	●	●		●
			●	●
				●
●	●			●
				●
				●
●	●	●	●	●
			●	●
●	●	●		●
		●		
				●
		●		●
				●
●				●
●	●			
●	●	●	●	●
				●
			●	
				●
				●
				●
				●
●				
●				
			●	●
			●	●

Spreadsheets

	VISCALC ADVANCED VERSION	SUPERCALC^2	MULTIPLAN
การนับจำนวน PAGE	●		
กำหนดขนาด PAGE L = ยาว W = กว้าง	L, W	L, W	L, W
SET MARGIN L = ซ้าย R = ขวา	L, R		L, R
SET MARGIN T = บน B = ล่าง			T, B
พิมพ์รายงานโดยให้ มี,หรือไม่มีขอบของ		●	●
แผ่นงานความสามารถพิเศษในการพิมพ์	●	●	●
เก็บ PRINT OPTION ที่ได้เลือกไว้แล้วและใช้งานโดยอัตโนมัติ	With keyboard macros		●
กำหนดขนาด speard sheet ให้พอดีกับกระดาษที่ใช้พิมพ์เองโดยอัตโนมัติ			
Single-sheet feed		●	
กำหนดไม่ให้พิมพ์ ตัวเลขที่ไม่ใช่ค่าบวกในเซล			
Specify no-print for models with nonpositive values in cells			
กำหนด REPORT TITLE ในการพิมพ์			
ไม่สนใจกับแถวที่ศูนย์			
ไม่พิมพ์ค่าที่เป็นศูนย์			
แสดงค่าศูนย์ + ด้วย –			
แสดงค่าศูนย์ เป็น –n,n– หรือ (n)			
ขนาดของเซล			
ปรับค่าได้	●	●	●
ความกว้างต่ำสุด	1	0 (removes column from view)	3
ความกว้างสูงสุด	125	127	32
Default	9	9	10
Calendar functions	เปลี่ยน CALENDAR DATE ให้เป็น ABSOLUTE DATE; เปลี่ยนชั่วโมง/นาที/วินาที ให้เป็นเศษส่วนของวัน	อ่านวันเดือนปีลงในเซลใส่วันเดือนปีที่กำหนด; แสดงจำนวนวันในสัปดาห์ และ JULIAN DATE	ไม่มี
Partial save/load to disk		●	Not directly—with file linking, external copies
เก็บข้อมูลเป็น TEXT FILE	●	●	●
ข้อจำกัด	คำสั่งส่วนมากทำงานที่ global level หรือเป็นคำสั่งเฉพาะ เซล ไม่สามารถ FORMAT area คอลัมน์ด้วยคำสั่ง FORMAT ได้	จำเป็นต้องมี buit-in funtion มากกว่านี้โดยเฉพาะ financial	Requires two-character ID for each row and column that requires more work on the part of the user
คู่มือใช้งาน	ดีมาก;มี TUTORIAL REFEREACE QUICK REFERENCE GUIDES COMMAND CHART	ดี	ดีมาก
Ease of use	Great number of commands and subcommands are difficult to remember; has useful on-screen "help" facility; simple in concept and execution	Has excellent "help" facility that is specific and relates to what is being done when called; 12 tutorial lessons are supplied on-disk	Makes very good use of special function keys; uses English-language menus that reduce learning time; expanded on-screen "help" facility; tutorial disk is included
Other distinctions	Unsurpassed cell attributes and good report formatting	Sorting capabilities for both rows and columns; easy to consolidate like worksheets; handles partial worksheets nicely	Color is beautifully supported; text can be used to a limited degree in formulas and functions; excellent table lookup feature; range names can be used in all formulas and functions; relative references to rows and columns are supported

MAGICALC	PERFECT CALC	PROCALC	TARGET: FINANCIAL MODELING	MICROPLAN
				●
L,W	W	L,W	L,W	L,W
L,R			L,R	L,R
T,B			T,B	T,B
				●
●		●	●	●
●		●	●	●
●	●			
●		●	●	
			●	
			●	
				●
				●
				●
				●
				●
●	●	●	●	●
0	0	0	2	4
36	76	Computer width array	50	20
7	9	12	9	9
ไม่มี	ไม่มี	System date or entered date function; system time or entered time lookup	เป็นฟังก์ชั่นที่แสดงวันเดือนปีปัจจุบัน	ไม่มี
			●	
●	●	●	●	●
ไม่เหมาะสำหรับงานที่ซับซ้อนใช้เมนูทำให้ทำงานช้า	Uses unfamiliar conventions to no advantage; uses unfamiliar column headings to no advantage	ไม่สามารถ INTERFACE กับระบบ GRAPHIC หรือ FILE MANAGER	ต้องการการวางแผนก่อนที่จะใช้งานมากกว่า SPREADSHEET อื่น	ไม่ flexible เหมือนกัน spreadsheet ส่วนใหญ่
ประกอบด้วย TUTORIAL SECTION REFERENCE GUIDE และ QUICK REFERENCE CARD	ประกอบด้วย KEY BOARD TEMPLATE และ QUICK REFERENCE GUIDE	ไม่ดีนัก	ไม่ดีนัก	ดีการพิมพ์ทำให้อ่านได้ง่าย TUTORIAL EXAMPLE ดีมาก
Very easy to get into and use, especially when using "Magic Window" with spreadsheet program; menu-driven system	Comparatively difficult to use owing to limited capabilities; 8-lesson tutorial diskette and prewritten templates are supplied	Tutorial is quite good; on-screen "help" function gives assistance by subject; comparatively easy to use	Good on-screen "help" capability; 106 keywords are provided to access help screens	Very easy to use; provides fully designed worksheet format that lets user simply fill in the blanks
Menus are different from those used in other systems that consist of a series of subsystems; no need to learn commands for difficult activities like file handling and printing	Time to learn system use is reduced if user is familiar with other Perfect software packages	System incorporates a text processor (UCSD Pascal editor)	Modeling can be done rapidly if user is familiar with other similar modeling systems	Interesting system with fixed format; has great many built-in functions

โดย สุวัฒน์ ปุณณชัย
ศิริภรณ์ บวรวนิชยกูร



WordStar ฉบับชาวบ้าน

ตอนจบ

คู่มือการใช้ Wordstar ตอนนี้จะเป็นตอนสุดท้ายแล้ว เมื่อจบตอนนี้แล้วหวังว่าผู้อ่านคงจะเข้าใจการใช้งาน Wordstar และใช้ได้อย่างมีประโยชน์มากที่สุด ในตอนนี้เราจะกล่าวถึงการตั้งเครื่องหมาย หรือ place marker เพื่อเป็นการบอก Wordstar ในการค้นหาใน file และการหาคำใน file เพื่อแก้ไข, เปลี่ยนแปลงเป็นคำอื่น ๆ

- การตั้ง Place marker

place marker คือตัวเลขตั้งแต่ 0 ถึง 9 ใช้เป็นเครื่องหมายกำหนดลงใน file ในส่วนที่เราต้องการเพื่อว่าเราจะกลับมายังตำแหน่งนั้น ๆ อีก การตั้ง place marker ใช้คำสั่ง CTRL K แล้วตามด้วยตัวเลขตั้งแต่ 0 ถึง 9 แล้วแต่เรากำหนดขึ้นมาเช่น CTRL KI CTRL K 2, CTRL K 3...เป็นต้น เมื่อเราใช้คำสั่งนี้แล้วบนจอตรงตำแหน่งที่ Cursor อยู่จะมี 1 หรือ 2 หรือ 3 ... ปรากฏขึ้นมาพอเรากำหนด-place marker เรียบร้อยแล้วในตำแหน่งนั้น ๆ เราจะเลื่อน Cursor ไปหาอีกครั้งทำได้ง่ายและรวดเร็วมาก โดยใช้คำสั่ง CTRL Q แล้วตามด้วยตัวเลขที่กำหนดไว้ Cursor ก็จะไปอยู่ที่ตำแหน่งนั้นทันที ยกตัวอย่างชัด ๆ เช่นใน file ของเรามีส่วนหนึ่งมีข้อความดังนี้

> NOTE : Good data processing procedure dictates that the user test the program, run and tesk sample sets of data, and run the system in operation of the programs are satis — factory.

เราต้องการที่จะตั้ง place marker ที่คำว่า Good เราจะต้องเลื่อน Cursor ไปที่ตำแหน่งหน้า คำว่า Good แล้วใช้คำสั่ง CTRL K 1 บนจอจะเป็นดังนี้

NOTE 1 Good data processing procedure dictates...

เมื่อ Cursor อยู่ตำแหน่งใด ๆ ก็ตาม ถ้าเราต้องการให้มาอยู่ที่ตำแหน่งตรงคำว่า Good เราจะใช้คำสั่ง CTRL Q 1 Cursor ก็จะมาปรากฏตรงตำแหน่งนั้นตามต้องการ

- การหาคำใน file

ใน Wordstar จะมีคำสั่งในการหาคำหรือวลีที่มีความยาวจนถึง 30 ตัวอักษรได้โดยใช้คำสั่งง่าย ๆ คือ CTRL Q F เมื่อเราสั่ง CTRL Q F แล้ว WordStar จะถามดังนี้ FIND? แล้วเราก็คีย์คำที่เราต้องการลงไป เสร็จแล้วกดคีย์ ESC Cursor ก็จะไปปรากฏอยู่ที่คำ ๆ ที่เราต้องการหาทันที ถ้าใน file นั้นมีคำที่เราหาอยู่หลายคำ คำที่ Cursor ไปปรากฏอยู่จะเป็นคำแรกที่ Wordstar หาเจอถ้าเป็นคำที่เราต้องการเราก็สามารถแก้ไขหรือทำอะไรต่อก็ได้ แต่ถ้าไม่ใช่เราจะใช้คำสั่ง CTRL L เพื่อให้ Wordstar หาคำ ๆ นั้นต่อไปใน file จนเมื่อ Wordstar ไม่พบก็จะบอกมาที่จอดังนี้ * * * NOT FOUND* * * คำที่หา * * * Press ESC Key * * * แสดงว่าคำนั้นไม่มีใน file แล้ว ถ้าเราต้องการจะกลับไปยังคำสุดท้ายที่หาเจอ เราจะใช้คำสั่ง CTRL Q V

ทีนี้การหาคำบอกคำเช่นคำว่า Under กับ Underline-WordStar ก็มีทางเลือกให้เรา ในตอนที่เราใช้คำสั่ง CTRL Q F แล้ว Wordstar ถามว่า FIND ? อะไรเมื่อเราคีย์คำที่ต้องการหาเสร็จแทนที่เราจะกด ESC คีย์เรากด RETURN แทน Wordstar จะถามต่อว่า Option ? อีก Option ที่ว่านี้มีอยู่ 3 Option ดังนี้

● Option U

Option U จะเป็นการบอก Wordstar ว่า คำที่หานั้นเป็นอักษรแบบตัวพิมพ์ใหญ่ (Upper Case) หรือ ตัวพิมพ์เล็ก-(Lower Case) ก็ได้

● Option W

Option W จะเป็นการบอก Wordstar ว่าให้หาคำนั้น ๆ ทั้งคำอย่างเดียว เช่น หาคำว่า Under ถ้าเราใช้ Option W นี้ แล้ว WordStar จะหาคำว่า Under เท่านั้น แต่คำว่า Underline หรือ Underscore----- ฯลฯ Wordstar จะไม่สนใจเลย

Option ทั้งสองนี้เราอาจจะใช้พร้อม ๆ กันก็ได้ คือ

Option WU หรือ Option UW ก็ได้

หลังจากการหาคำใน file แล้ว ทีนี้เราจะหาคำนั้น ๆ แล้วแทนที่ด้วยคำอื่น ๆ ด้วย Wordstar มีคำสั่งที่ให้หาคำแล้วแทนที่คำนั้นด้วยคำอื่นเลย เราจะใช้คำสั่ง CTRL Q A ยกตัวอย่างเช่น เราต้องการหาคำว่า Word เราใช้คำสั่ง CTRL Q A แล้ว WordStar จะถามว่า FIND ? เราคีย์คำว่า Word ไปพร้อมทั้ง RETURN Wordstar จะถามต่อว่า REPLACE WITH? ตอนนี้เราต้องการแทนคำว่า Word ด้วยคำว่าอะไร เราก็คีย์ลงไป แล้วกดคีย์ ESC เมื่อ Wordstar หาคำว่า Word เจอแล้ว ก็จะถามเราอีกว่า REPLACE (Y/N) ทีนี้ถ้าเราตอบ y ก็จะเป็นการอนุญาตให้แทนที่ได้ แต่ถ้าคำว่า Word นั้น ไม่ใช่คำว่า Word ที่เราต้องการ เราจะกด CTRL L เพื่อให้ Word Star หาคำว่า Word ตัวอื่นที่ถัดไปอีก การหาคำแล้วแทนที่อย่างนี้ บางครั้งก็ยุ่งยากเพราะในการแทนที่คำซ้ำ ๆ กันเราจะต้องทำเองทุกคำใน file แต่ถ้าเรามีความแน่ใจว่าคำ ๆ นั้นทุกคำใน file จะต้องถูกเปลี่ยนเป็นคำอื่น เราก็มีวิธีที่จะให้ Wordstar ทำได้โดยอัตโนมัติ ดังนี้

● Option N

Option N นี้ ใช้สำหรับการหาคำแล้วแทนที่คำนั้นด้วยคำอื่นอย่างอัตโนมัติ เช่นเราต้องการแทนที่คำ Word

ทุก ๆ คำใน file ด้วย คำว่า number ตอนแรกเราใช้คำสั่ง CTRL Q A เหมือนเดิม Wordstar จะถามว่า FIND ? แล้วเราคีย์คำว่า Word พร้อมทั้งกด RETURN ต่อมา Wordstar ก็จะถามว่า REPLACE WITH ? เราก็คีย์คำว่า number พร้อมทั้ง RETURN สุดท้าย Wordstar จะถาม OPTIONS? เราใช้ N แล้วกด RETURN Wordstar จะแทนที่คำว่า Word ใน file ทุก ๆ คำด้วย คำว่า number โดยอัตโนมัติ

มาถึงตอนนี้เราก็เรียนรู้การหาคำและแทนที่คำด้วยคำอื่นแล้ว และการตั้ง marker เพื่อการหาตำแหน่งนั้นใน file ต่อไปเราจะกล่าวถึงการจัดการเกี่ยวกับ block ข้อมูล และ file ดังต่อไปนี้ ในตอนแรก ๆ เราก็พอจะรู้บ้างแล้วถึงการ save file ของเราในตอนนี้เราจะกล่าวกันอย่างละเอียดอีกครั้ง

● การ save file แบบชั่วคราว คือในขณะที่เรากำลังพิมพ์ข้อความอะไรอยู่ก็ตามแต่ยังไม่เสร็จและเราจะ save file นี้ไว้ก่อน เมื่อ save เสร็จแล้วเราจะใช้ file นั้นต่อเราจะใช้คำสั่ง CTRL K S คำสั่งนี้จะมีประโยชน์มากเช่นขณะที่เราคีย์ข้อความลงใน file อาจจะเกิดไฟดับ หรืออุบัติเหตุอะไรก็ตาม เราก็ยังมีส่วนของ file นั้น save ไว้อยู่บางส่วน

● การ save file แล้วปิด file นั้นเลย คือเรา save file นั้นแล้ว และจะทำอะไรอย่างอื่นต่อไปอีก เช่น เปิด file ใหม่ หรือพิมพ์ file เป็นต้น เราจะใช้คำสั่ง CTRL K D เมื่อเราใช้คำสั่งนี้แล้ว Wordstar จะทำการ save file แล้วเข้าสู่ no - file menu เลย

● การ save file แล้วออกจากการใช้งาน Wordstar การ save แบบนี้คล้ายกับแบบ CTRL K D แต่ แทนที่ Wordstar จะเข้าสู่ no-file menu Wordstar จะหยุดการทำงานและเข้าสู่ CP/M เลย เราจะใช้คำสั่ง CTRL K X

● การยกเลิก file เราจะทำการยกเลิก file คือไม่ save file นั้น ๆ ลงไปใน disk ด้วยคำสั่ง CTRL K Q คำสั่งนี้ใช้ในกรณีที่เราทำการแก้ไข file ผิดไปมากและเราไม่ต้องการใช้ หรือเราเปิด file ผิด file เป็นต้น

● การย้ายตำแหน่งของ Block โดยก่อนอื่นต้องเลื่อน CURSOR ไปไว้ในตำแหน่งเริ่มต้นของ Block ใช้คำสั่ง CTRL XB บนจอจะแสดง B ให้เห็น หลังจากนั้นเลื่อน CURSOR ไปยังตำแหน่งสุดท้ายของ Block ใช้คำสั่ง CTRL KK บนจอจะเห็นเป็น X ปรากฏ เท่านี้ก็เป็นการบอกตำแหน่งเริ่มต้นและจุดจบของ Block ให้ Wordstar รู้ เมื่อต้องการย้าย Block นั้นไว้ ณ. ตำแหน่งอื่นใน file ก็ให้ย้าย CURSOR ไปไว้ในตำแหน่งที่ต้องการแล้วใช้คำสั่ง CTRL KV Block นั้นทั้ง Block ก็จะปรากฏแทรก ณ. ตำแหน่งใหม่นั้น

● การ COPY BLOCK บางทีข้อความบางข้อความใน file หนึ่ง ๆ อาจจะปรากฏใน file มากกว่า 1 ครั้ง ดังนั้นการ COPY BLOCK จะช่วยให้สะดวกขึ้นมาก เราจะใช้คำสั่ง CTRL K C BLOCK ซึ่งเรากำหนดจุดเริ่มต้นและจุดจบไว้แล้วก็จะปรากฏในอีกส่วนของ file โดยที่จะไม่หายไปจากตำแหน่งเดิมด้วยการ COPY BLOCK หนึ่ง ๆ อาจทำได้มากกว่า 1 ครั้ง เพราะตำแหน่งที่เรา mark เอาไว้จะไม่หายไปไหนจนกว่าเราจะ SAVE FILE นั้นแล้ว หรือเปลี่ยนตำแหน่งเริ่มต้นและ

จบใหม่ บางครั้งถ้าเราไม่ต้องการให้ตำแหน่งของ BLOCK ปรากฏบนจอให้ใช้คำสั่ง CTRL K H เพื่อซ่อนตำแหน่งนั้นไว้

● การลบ BLOCK ที่ไม่ต้องการออกจาก file ใช้คำสั่ง CTRL KY BLOCK ที่เรากำหนดตำแหน่งไว้ก็จะถูกลบไปโดยไม่คำนึงถึงตำแหน่งของ CURSOR ในขณะนั้น บางครั้งตำแหน่งของ BLOCK ที่กำหนดไว้อาจจะซ่อนอยู่ด้วยคำสั่ง CTRL KH ดังกล่าวเพื่อดูตำแหน่งของ BLOCK ให้แน่ใจให้ใช้คำสั่ง CTRL KH อีกครั้ง ตำแหน่งของ BLOCK ก็จะปรากฏบนจอตามเดิม หลังจากคำสั่ง CTRL KY แล้วให้ใช้คำสั่ง CTRL KQ เพื่อยกเลิก file นั้น

● การย้าย Block จาก file หนึ่งไปยังอีก file หนึ่ง หลังจากกำหนดตำแหน่ง Block แล้วใช้คำสั่ง CTRL K W Wordstar จะถามว่า NAME OF FILE TO WRITE MARKED TEXT ON ? เราจะต้องป้อนชื่อ file ที่เราจะย้าย Block ไปไว้เป็นการชั่วคราว สมมุติเราต้องการ file X ก็ให้พิมพ์ X แล้วกดแป้น RETURN ดังนั้นใน file X ของเราก็จะประกอบด้วยข้อความใน Block ที่เราย้ายมาจาก file เก่า แล้วใช้คำสั่ง CTRL KQ เพื่อยกเลิก file นั้น

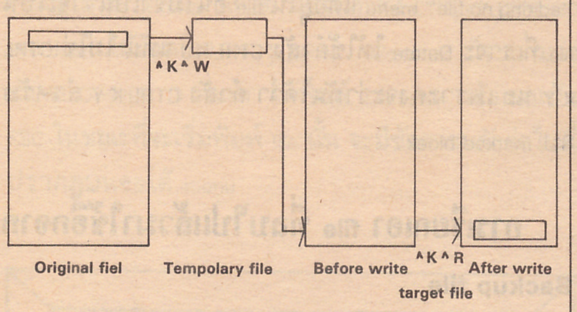
การอ่านข้อความใน file หนึ่งส่งผ่านไปยัง file อื่น ในหัวข้อก่อนเราเพียงแต่เขียนข้อความใน Block จาก file ที่เปิดใหม่ คราวนี้เราจะอ่าน file ใหม่นั้นแล้วเก็บไว้ในอีก-file หนึ่ง ซึ่งเคยถูกเปิดมาก่อนแล้ว ก่อนอื่นจะต้องเลื่อน CURSOR ให้มาอยู่ระหว่างบรรทัดที่เราจะแทรกข้อความ แล้วใช้คำสั่ง CTRL K R Wordstar จะถามว่า NAME OF FILE TO READ? ให้พิมพ์ X แล้วกดแป้น Return Wordstar ก็จะอ่านข้อความทั้งหมดจาก file X มาเก็บไว้ใน file ที่เราต้องการ แล้วใช้คำสั่ง CTRL KQ เพื่อยกเลิก file นั้น

สรุปการย้ายตำแหน่งของ Block จาก file หนึ่งไปอีก file หนึ่ง ผู้อ่านจะได้เข้าใจยิ่งขึ้น สรุปได้เป็น 2 ข้อ คือ

1. บอกตำแหน่งของ Block ที่จะย้ายจาก file เริ่มต้นก่อนแล้วเขียนลงใน file ที่สร้างขึ้นชั่วคราว

2. เปิด file ที่เราต้องการให้ Block นั้นมาอยู่ เลื่อน cursor ให้อยู่ในตำแหน่งที่ต้องการแล้วอ่าน Block นั้นจาก file ที่เปิดชั่วคราวนั้นมาไว้

เท่านี้การ copy Block จาก file เริ่มต้นไปยัง file ที่ต้องการโดยผ่านทาง file ซึ่งคราวที่สร้างขึ้นก็เป็นอันเสร็จ แสดงได้ดังรูปข้างล่างนี้



maing text from are file to anothor

● การ COPY FILE เมื่อเราเข้าสู่ Wordstar แล้วอยู่ใน-"editing no file" menu ใช้คำสั่ง O เพื่อบอก Wordstar ว่าเราจะทำการ COPY FILE Wordstar จะถามว่า NAME OF FILE TO COPY FROM ? สมมุติว่าเราต้องการ file x ก็ให้ป้อนชื่อ x เข้าไปแล้วกด RETURN คราวนี้ Wordstar ก็จะถามอีกว่า NAME OF FILE TO COPY TO? ให้ป้อนชื่อ FILE ที่ต้องการเข้าไปเท่านี้ Wordstar ก็จะทำการ COPY FILE X ลงใน FILE ที่เราต้องการ แต่ถ้าหากเราไม่ได้อยู่ใน "editing no file" แต่อาจจะอยู่ใน file อื่นให้ใช้คำสั่ง CTRL KO แทน ก็จะได้ผลเช่นเดียวกับ

● การตั้งชื่อ FILE ใหม่ ถ้าหากอยู่ใน "editing no file" memu ใช้คำสั่ง E Wordstar จะถาม NAME OF FILE TO RENAME? สมมุติให้เป็น X พิมพ์ X กด Return คราวนี้ก็จะปรากฏ NEW NAME? แทน พิมพ์ Z แล้วกด Return คราวนี้ FILE X ก็จะไม่ปรากฏใน directory แต่จะมี FILE ชื่อ Z แทน ซึ่งจะปรากฏอยู่ใต้ "editing no file" menu คราวนี้ให้เก็บ COPY ของ FILE นี้ไว้ใน DISK โดย กดแป้น D (เพื่อเปิด document file Wordstar ก็จะถาม "NAME OF FILE TO EDIT? กด Z แล้ว Return เป็นอันว่า file ใหม่ชื่อ Z ถูกเก็บไว้ใน DISK เรียบร้อย

ถ้าหากว่าต้องการ RENAME FILE ในขณะที่อยู่ ใน file ใด ๆ โดยไม่จำเป็นต้องอยู่ใน file ที่จะ RENAME ให้ใช้คำสั่ง CTRL KE แทน

● การลบ file ให้กด y ถ้าอยู่ใน "editing no file" menu Wordstar จะถาม NAME OF FILE TO DELEIE? พิมพ์ Z แล้วกด Return Wordstar ก็จะทำการลบ file Z ออกไปซึ่ง FILE Z ก็จะไม่ปรากฏใน directory แต่ถ้าหากว่าไม่ได้อยู่ใน "editing no file" menu แต่อยู่ใน file อื่นไม่จำเป็นว่าจะเป็น file ที่เราจะ Detete ให้ใช้คำสั่ง CTRL K J แทน ไม่ใช่ CTRL K Y นะ เพราะคงจะว่ากันได้ว่า คำสั่ง CTRL K Y สำหรับลบ marked block

การเรียกเอา file ที่ลบไปแล้วมาใช้อีกจาก

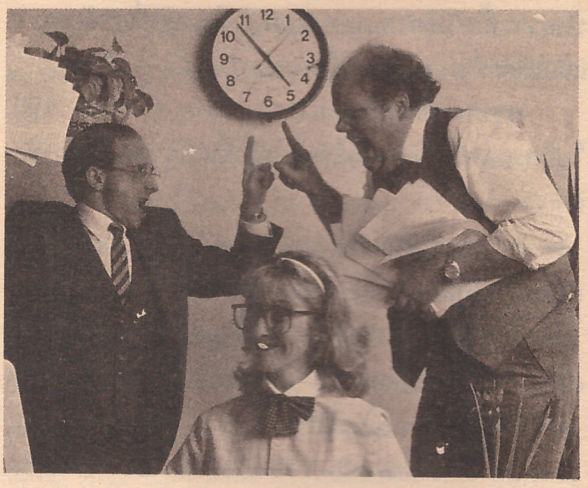
Backup file

ทีนี้ถ้าหากว่าเราลบ file Z ไปแล้วแต่ความจริงยังต้องการใช้ FILE Z อยู่ เราสามารถเก็บ file Z ใหม่ได้จาก "Z. BAK" โดยเมื่อ "editing no Sile" menu ปรากฏบนจอให้กดแป้น O Wordstar จะถาม "NAME OF FILE TO COPY FROM?" ให้พิมพ์ **Z.BAK** แล้วกด RETURN Wordstar จะถามอีกว่า "NAME OF FILE TO COPY TO?" คราวนี้พิมพ์ Z แล้ว RETURN Wordstar ก็จะทำการ COPY Z.BAK (ซึ่งก็คือ file ที่พึ่งลบไปจาก disk แต่ยังคงเก็บอยู่ใน memory) ไปเก็บไว้ใน file Z ใน disk ดังนั้น file Z ก็จะปรากฏบน directory ให้เห็น

คราวนี้มาพูดกันถึงวิธีการพิมพ์ document ที่ยาวกว่า 1 หน้าบ้าง ซึ่งจะต้องมีวิธีที่จะบอก Wordstar ว่าจะให้พิมพ์ถึงไหนขึ้นหน้าใหม่เมื่อไหร่ และจะพิมพ์แต่ละหน้าในรูปแบบใด

● การตั้งเงื่อนไขให้ Wordstar รู้ว่าจะขึ้นหน้าใหม่เมื่อไร เมื่อเราต้องการพิมพ์ document file ที่ยาวกว่า 1 หน้า และแต่ละหน้ามีรูปแบบที่แตกต่างกันไป เราจะต้องตั้งเงื่อนไขให้ Wordstar รู้ว่าจะต้องหยุดพิมพ์ที่ไหนสำหรับแต่ละหน้า ถ้าหากไม่บอก Wordstar ก็จะไม่รู้ว่าจะต้องหยุดพิมพ์ตรงไหนสำหรับหน้านี้ดังนั้นก็จะหยุดพิมพ์และขึ้นหน้าใหม่เองโดยอัตโนมัติ ในตำแหน่งที่เราไม่ต้องการ เมื่ออยู่ใน "editing no file menu ให้เปิด document file โดยกดแป้น d Wordstar จะถามว่า NAME OF FILE TO EDIT? สมมุติให้ file ที่เราต้องการเปิดชื่อ special ให้พิมพ์ special แล้วกดแป้น RETURN เมื่อ file special ปรากฏบนจอให้เลื่อน CURSOR ไปยังบรรทัดที่อยู่เหนือ paragraph ที่เรากำลังจะกำหนดตำแหน่งขึ้นหน้าใหม่ให้ Wordstar รู้ ตรวจดูที่มุมบนขวาว่า INSERT On อยู่หรือไม่ถ้าไม่มีใช้คำสั่ง CTRL V แล้ว RETURN เพื่อสร้างบรรทัดว่างใหม่ขึ้นอีกบรรทัด พิมพ์ **.cp 5** แล้วกดแป้น RETURN หลังจากนั้นให้ save copy file นั้น โดยใช้คำสั่ง CTRL KD คำสั่ง **.CP** เป็นเงื่อนไขที่จะบอกให้ Wordstar รู้จำนวนของบรรทัดที่จะพิมพ์ (ในที่นี้ 5 บรรทัด) ซึ่งสั่งให้ Wordstar พิมพ์ 5 บรรทัดต่อไปด้วยกัน ถ้าหากว่าพิมพ์ในหน้าแรกไม่พอให้ยกไปพิมพ์ในหน้าต่อไปรวมกันทั้ง 5 บรรทัดเลย ลองสั่ง PRINT ออกมาทาง PRINTER ดูโดยใช้คำสั่ง P และ ESCAPE

● การพิมพ์หัวข้อที่มุมบนสุดของหนังสือ ธรรมดาในการพิมพ์หนังสือ มักจะมีชื่อหนังสือ ชื่อบทที่มุมบนสุดของหนังสือทุกหน้า เพื่อให้ Wordstar รู้ว่าจะต้องพิมพ์ในรูปแบบเช่นนี้ ก่อนอื่นให้เปิด document file คือ file ชื่อ Datastar ตรวจดูว่า INSERT-ON อยู่หรือไม่ ถ้าไม่ใช้คำสั่ง CTRL V เพื่อให้ INSERT-ON ได้ แล้วใช้คำสั่ง CTRL N เพื่อสร้างบรรทัดว่างขึ้นมาใหม่ 1 บรรทัด พิมพ์ **.he ^S Micropro Brochure ^S** แล้ว Return (ต้องใช้คำสั่ง CTRL PS เพื่อให้เข้าถึงคำสั่ง CTRL S ในที่นี้ซึ่งเป็นคำสั่งให้ขีดเส้นใต้แต่ละอักษรตั้งแต่ตัว M ในคำว่า Micropro ถึงตัว E ในคำว่า Brochure) ดังนั้น ในหนังสือแต่ละหน้าที่จะพิมพ์ก็จะมีชื่อหนังสือทางมุมบนสุดแล้ว หัวข้อก่อนนี้เราได้รู้ถึงวิธีการพิมพ์หน้าหนังสือ



ตรงกลางหน้ากระดาษด้านล่างแล้ว ต่อไปถ้าต้องการให้พิมพ์หน้ากระดาษตรงมุมล่างสุดริมกระดาษด้านนอกในบรรทัดต่อไป จะต้องพิมพ์ .of blank แล้วใช้คำสั่ง CTRL PK เลื่อน cursor ไปจนถึง column ที่ 69 แล้วพิมพ์ # เพื่อบอกหน้า save file โดยใช้คำสั่ง CTRL KD

คำสั่ง .he นั้นเป็นคำสั่งที่ให้พิมพ์หัวข้อหนังสือในแต่ละหน้าได้ และคำสั่ง fo เป็นคำสั่งที่บังคับให้ Wordstar พิมพ์หน้ากระดาษที่มุมริมล่างสุด ลองพิมพ์ออกทาง PRINTER ดูก็จะเห็น

● การพิมพ์เพียงบางหน้า เราสามารถพิมพ์เพียงบางหน้าของหนังสือได้ เช่น ถ้าเราต้องการพิมพ์เฉพาะหน้า 2-3 ของ file Datastar เมื่ออยู่ใน "editing no file" menu ให้กด P Wordstar ก็จะถาม NAME OF FILE TO PRINT ? พิมพ์ Datastar แล้ว RETURN Wordstar ก็จะถามว่า START AT PAGE ? พิมพ์ 2 RETURN Wordstar ก็จะถามอีกว่า STOP AFTER PAGE NUMBER (RETURN FOR END) ? พิมพ์ 3 แล้วจึง RETURN ให้กดแป้น RETURN อีก 4 ครั้ง เพื่อข้ามคำถามที่เหลือไป และเริ่มพิมพ์ออกทาง printer ลองพิมพ์ดูจะเห็นว่า Wordstar จะเริ่มพิมพ์ในหน้า 2 และจบในหน้า 3

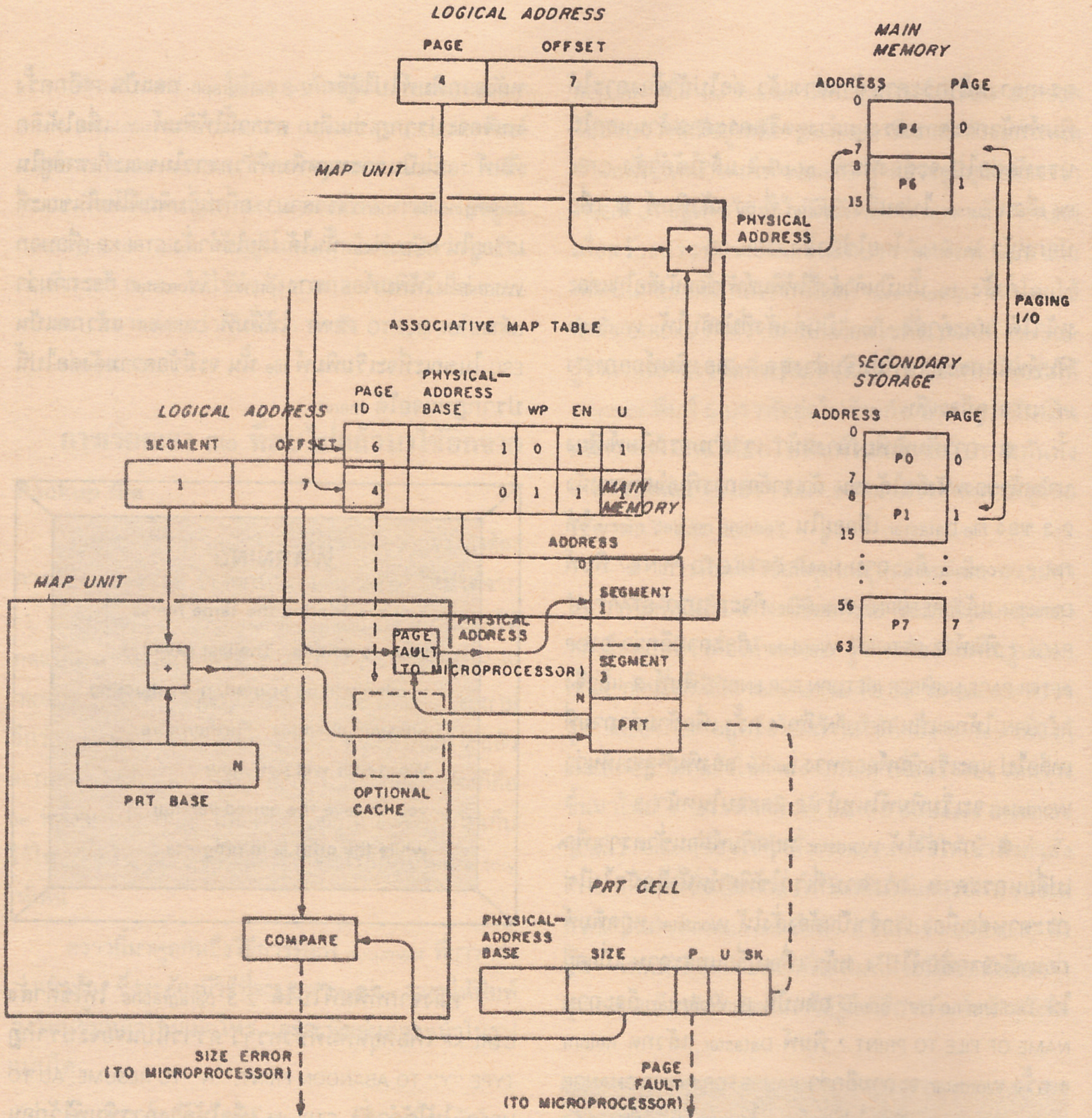
● การสั่งให้ Wordstar หยุดพิมพ์ก่อนชั่วคราวเพื่อเปลี่ยนกระดาษ กระดาษที่เราใช้พิมพ์หนังสือมักไม่ใช่กระดาษต่อเนื่อง เราจำเป็นต้องสั่งให้ Wordstar หยุดพิมพ์ก่อนหลังจากพิมพ์ไป 1 หน้า เพื่อเปลี่ยนกระดาษ เมื่ออยู่ใน "editing no file" menu กดแป้น P Wordstar ก็จะถาม NAME OF FILE TO PRINT ? พิมพ์ Datastar แล้วกด Return 4 ครั้ง Wordstar จะถามอีกว่า PAUSE FOR PAPER CHANGE BETWEEN PAGES (Y/N) ? ให้พิมพ์ Y แล้ว RETURN อีก 4 ครั้ง เพื่อเริ่มพิมพ์ หลังจากที่พิมพ์ไปได้ 1 หน้าแล้ว PRINTER ก็จะหยุดพิมพ์ก่อนเพื่อให้เปลี่ยนกระดาษ เมื่อเปลี่ยนเรียบร้อยแล้วต้องการพิมพ์ต่อก็ให้กดแป้น P PRINTER ก็จะพิมพ์ต่อไป

เมื่ออยู่ใน "editing no file" menu กดแป้น P Wordstar จะถาม NAME OF FILE TO PRINT ? พิมพ์ Datastar แล้วกดแป้น ESC หลังจากที่ได้พิมพ์ไป 2-3 paragraphs ให้พิมพ์ P อีกครั้ง บนจอก็จะปรากฏ TYPE "Y" TO ABANDON PRINT, "N" TO RESUME, U TO HOLD ให้พิมพ์ N เพื่อให้พิมพ์ต่อไปอย่างเก่า หลังจากพิมพ์ไปได้อีก 2-3 paragraph กดแป้น P อีกครั้ง บนจอจะปรากฏเช่นเดิม คราวนี้ให้พิมพ์ Y เพื่อให้เลิกพิมพ์ นี่เป็นการหยุดพิมพ์ชั่วคราวในขณะที่เราอยู่ใน "editing no file" menu เราสามารถที่จะเริ่มพิมพ์ใหม่ในขณะที่เราอยู่ใน file Datastar นั้นได้ โดยใช้คำสั่ง CTRL KP เพื่อบอก Wordstar ให้พิมพ์ออกทาง printer Wordstar ก็จะถามว่า NAME OF FILE TO PRINT ให้พิมพ์ Datastar แล้วกดแป้น ESC ในขณะที่จะเริ่มพิมพ์ file นั้น จะมีข้อความดังต่อไปนี้ปรากฏบนจอใต้ menu

WARNING

You are printing the same file as you are editing. The last saved version will be printed, not reflecting unsaved changes. Furthermore, Word Star will not allow you to save the edited version while the print is in progress.

หลังจากที่พิมพ์ไปได้ 2-3 paragraphs ให้ใช้คำสั่ง CTRL KP เพื่อหยุดพิมพ์ชั่วคราว คราวนี้บนจอจะปรากฏ TYPE "Y" TO ABANDON PRINT, "N" TO RESUME, AU TO HOLD ให้ใช้คำสั่ง CTRL U เพื่อให้ค้างการพิมพ์ไว้ก่อน คราวนี้บนจอก็จะปรากฏ "***INTERRUPTED*** PRESS ESCAPE KEY" ให้กดแป้น ESC ในระหว่างนี้เราสามารถเลื่อน cursor ไปยังตำแหน่งใดก็ได้ ลบหรือย้ายตำแหน่ง paragraph อย่างไรก็ได้ (จะไม่มีการเปลี่ยนแปลงเกิดขึ้นกับส่วนที่พิมพ์ไปแล้ว) เมื่อเรียบร้อยแล้ว ให้ใช้คำสั่ง CTRL KP เพื่อพิมพ์ต่อ เมื่อพิมพ์จนพอแล้วให้ใช้คำสั่ง CTRL KP เพื่อหยุดการพิมพ์อีกครั้ง บนจอก็จะปรากฏเช่นเดิม ให้กดแป้น Y เพื่อยกเลิกการพิมพ์ แล้วใช้คำสั่ง CTRL KG เพื่อยกเลิก file นั้น ■



Virtual Memory สำหรับ ไมโครคอมพิวเตอร์

โดยวิสาร กำจรเวทย์

เมื่อไม่นานมานี้ เมื่อระบบไมโครคอมพิวเตอร์ใดมีหน่วยความจำมากกว่า 32 กิโลไบท์ ก็นับได้ว่าเป็นระบบที่หรูหรามากพอควรทีเดียว เพราะเหตุว่าหน่วยความจำในขณะนั้นมีราคาแพงมาก ซึ่งจะทำให้คุณมีปัญหามากในการที่จะยัดเยียดโปรแกรมให้เข้าไปอยู่ในหน่วยความจำขนาดเล็กที่คุณมีอยู่ แต่เดี๋ยวนี้คุณสามารถซื้อหน่วยความจำที่มีขนาดถึง 256 กิโลไบท์ได้ในราคาต่ำกว่า 500 เหรียญสหรัฐ ทั้งยังมีระบบไมโครคอมพิวเตอร์ที่มีขนาด 16 บิทและ 32 บิทที่สามารถอ้างแอดเดรสในหน่วยความจำได้มากกว่าของระบบไมโครคอมพิวเตอร์ขนาด 8 บิทนับ 100 เท่า พร้อมทั้งยังมีการเปลี่ยนแปลงครั้งใหญ่ในโลกของไมโครคอมพิวเตอร์อีกด้านก็คือทางด้านซอฟท์แวร์ เช่นระบบ Multitasking Operating System, High Level Language และซอฟท์แวร์ทางธุรกิจที่ใช้งานสะดวก ก็กำลังเป็นที่นิยมกันมากขึ้น ดังนั้นปัญหาที่จะตามมาก็คือ ทำอย่างไรเราจึงจะสามารถใช้ประโยชน์จากการเปลี่ยนแปลงทั้งทางด้านฮาร์ดแวร์ที่มีประสิทธิภาพ และด้านซอฟท์แวร์ที่ซับซ้อนนี้ได้

เทคนิคการใช้ Virtual Memory ก็เป็นความพยายามที่จะใช้ประโยชน์จากความก้าวหน้าอันนี้ Virtual Memory *เป็นระบบอัตโนมัติที่ใช้สำหรับควบคุมหน่วยความจำขนาดใหญ่มาก ๆ* ดังจะเห็นว่ามีการผลิตชิ้นส่วนพิเศษทางฮาร์ดแวร์เพื่อใช้กับระบบนี้คือ ไอซีชิพเดียวที่ทำหน้าที่เป็น MMU (Memory Management Unit) ซึ่งได้พัฒนาขึ้นมาเพื่อที่จะให้เหมาะสมกับระบบไมโครคอมพิวเตอร์โดยเฉพาะ

ไมโครคอมพิวเตอร์แทบทุกระบบมีอุปกรณ์ที่ใช้เป็นหน่วยความจำหลาย ๆ แบบด้วยกัน และอุปกรณ์แต่ละชิ้นก็มีความจุและความเร็วในการทำงานที่ต่างกัน ตามหลักการพื้นฐานของเทคโนโลยี่เกี่ยวกับไมโครคอมพิวเตอร์แล้ว *ความเร็วในการทำงานของหน่วยความจำนั้นสัมพันธ์โดยตรงกับราคาของมัน* ถ้ามีขนาดใหญ่จะมีความเร็วต่ำและราคาไม่แพงเท่าไหร่ แต่ถ้าความเร็วสูงจะมีขนาดเล็กและราคาแพง ตัวอย่างทั่ว ๆ ไปก็คือ อุปกรณ์หน่วยความจำแบบดิสค์และแบบ RAM เมื่อนำไปใช้งาน การส่งถ่ายข้อมูลระหว่างทั้งสองประเภทจะต้องมีน้อยที่สุดมิฉะนั้นเวลาของการเข้าถึง (access) โดยส่วนรวมแล้วจะช้าเพราะเวลาของการเข้าถึงจะถูกกำหนดด้วยเวลาของการเข้าถึงอุปกรณ์ที่ช้ากว่า ในการใช้งานนั้นความเร็วของหน่วยความจำหลักซึ่งเป็น RAM จะเร็วกว่าหน่วยความจำสำรองเช่นดิสค์แม่เหล็กราว ๆ 1,000-5,000 เท่า ดังนั้นจะต้องมีการเข้าถึงดิสค์ให้น้อยที่สุดเท่าที่จะเป็นไปได้และพยายามใช้หน่วยความจำหลักซึ่งเร็วกว่าแทน

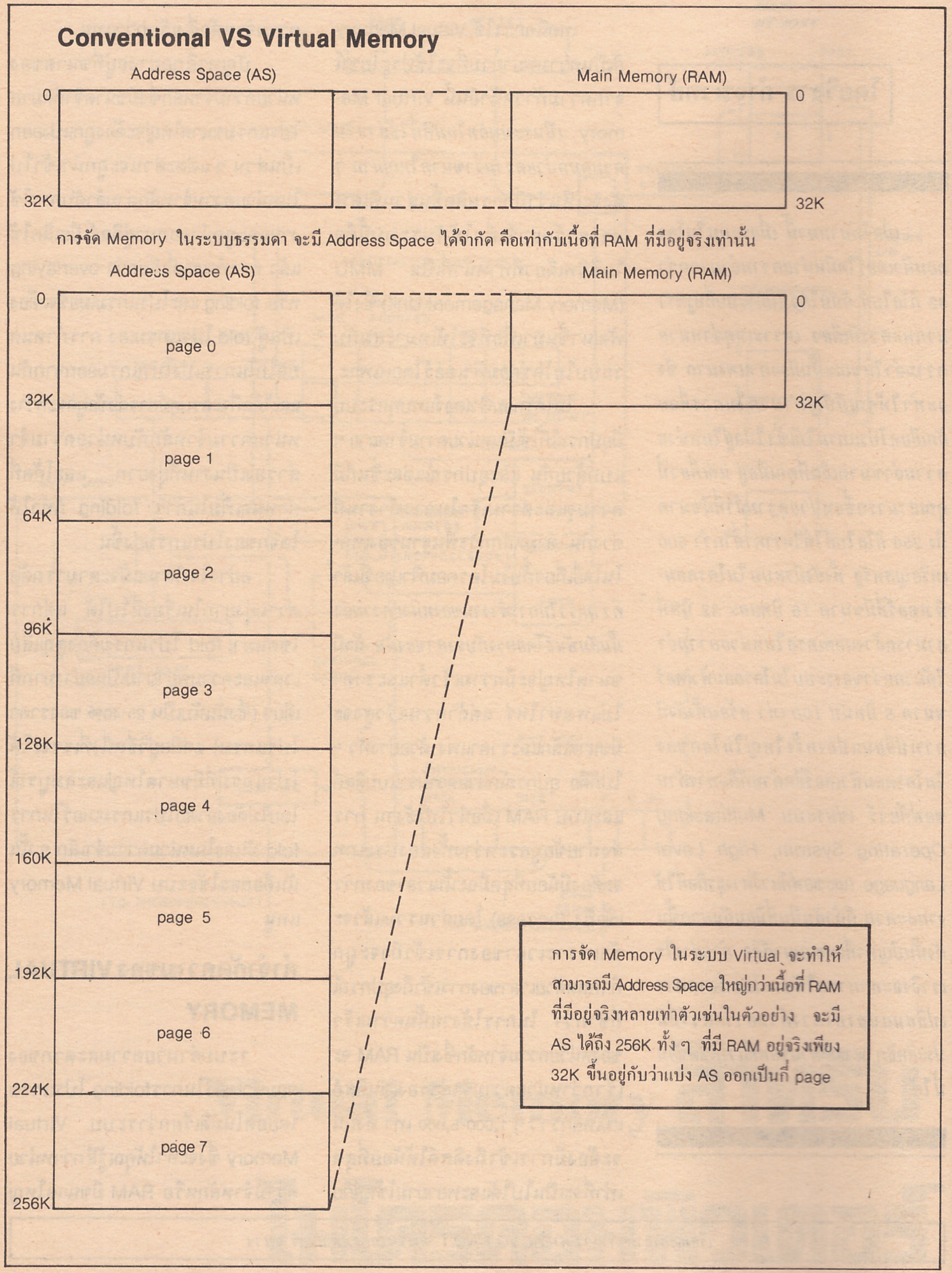
ปัญหาอีกอย่างอยู่ที่ขนาดของหน่วยความจำหลักซึ่งมีขนาดจำกัดมาก โปรแกรมขนาดใหญ่จะต้องถูกแบ่งออกเป็นส่วน ๆ แต่ละส่วนจะถูกนำเข้าไปในหน่วยความจำหลักตามลำดับการใช้งานและถูกนำออกมาสู่ดิสค์เมื่อเลิกใช้แล้ว การทำเช่นนี้เรียกว่า overlaying หรือ folding และโปรแกรมเมอร์จะต้องเป็นผู้ fold โปรแกรมเอง การกำหนดกลไกในการแบ่งโปรแกรมออกจากกันและโค้ดที่จะควบคุมการส่งข้อมูลระหว่างหน่วยความจำหลักกับหน่วยความจำสำรองเป็นงานที่ยุ่งยาก และโค้ดที่กำหนดเพิ่มในการ folding ก็ทำให้โลจิกของโปรแกรมยุ่งขึ้น

อย่างไรก็ตามแม้จะสามารถตัดความยุ่งยากในเรื่องนี้ไปได้ แต่การใช้คนมา fold โปรแกรมต้องสูญเสียเวลาและความพยายามเป็นอย่างมากทีเดียว (ซึ่งนั่นนับเป็น 25-40% ของราคาโปรแกรม) แต่มีอยู่วิธีหนึ่งที่เราจะได้โปรแกรมที่มีขนาดใหญ่และสมบูรณ์โดยไม่ต้องอาศัยโปรแกรมเมอร์ในการ fold มันลงในหน่วยความจำเล็ก ๆ นั้น นั่นคือการใช้ระบบ Virtual Memory แทน

คำจำกัดความของ VIRTUAL MEMORY

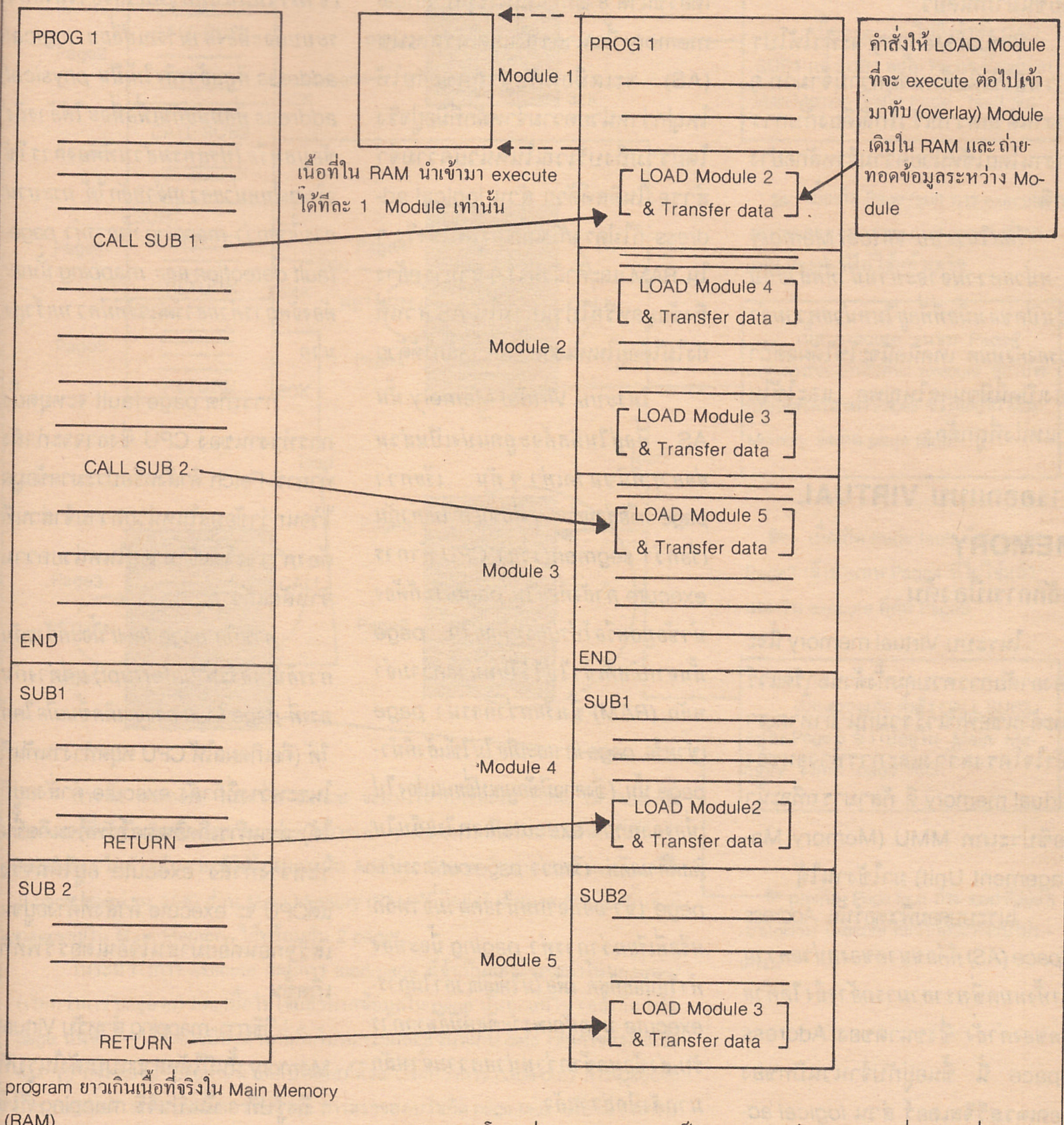
ระบบอำนวยความสะดวกของคอมพิวเตอร์ในการfolding โปรแกรมโดยอัตโนมัติเรียกว่าระบบ Virtual Memory ซึ่งจะทำให้คุณรู้สึกว่าหน่วยความจำหลักหรือ RAM มีขนาดใหญ่

เรียบเรียงจากวารสาร IDEAL VO.1 NO.1 ของชมรมคอมพิวเตอร์ จุฬาฯ


Conventional VS Virtual Memory
Address Space (AS)
Main Memory (RAM)
0
32K
0
32K
การจัด Memory ในระบบธรรมดา จะมี Address Space ได้จำกัด คือเท่ากับเนื้อที่ RAM ที่มีอยู่จริงเท่านั้น
Address Space (AS)
Main Memory (RAM)
0
page 0
32K
page 1
64K
page 2
96K
page 3
128K
page 4
160K
page 5
192K
page 6
224K
page 7
256K
0
32K
การจัด Memory ในระบบ Virtual จะทำให้สามารถมี Address Space ใหญ่กว่าเนื้อที่ RAM ที่มีอยู่จริงหลายเท่าตัวเช่นในตัวอย่าง จะมี AS ได้ถึง 256K ทั้ง ๆ ที่มี RAM อยู่จริงเพียง 32K ขึ้นอยู่กับว่าแบ่ง AS ออกเป็นกี่ page

รูปที่ 1

การ folding program



การ folding โดยแบ่ง program ออกเป็น Module ย่อย ๆ และเพิ่มเติมคำสั่งในการ LOAD และถ่ายทอดข้อมูลไปยังแต่ละ Module ตามจุดต่าง ๆ ที่จะต้องมีการทำ overlay ระหว่าง Module คำสั่งให้ LOAD Module ที่จะ execute ต่อไปเข้ามาทับ (overlay) Module เดิมใน RAM และถ่ายทอดข้อมูลระหว่าง Module

รูปที่ 2

กว่าที่เป็นจริงมากโดยไม่จำเป็นต้องมี Intermediate File, overlays และขบวนการเข้าถึงไฟล์ที่ยุ่งยากเหมือนแต่ก่อน ซึ่งทำให้โลจิกของตัวโปรแกรมง่ายขึ้นมากทีเดียว

Virtual Memory จะทำให้โปรแกรมที่ใช้เนื้อที่หน่วยความจำมาก ๆ ทำงานด้วยความเร็วใกล้เคียงกับการทำงานโดยใช้หน่วยความจำหลักอย่างปกติ

โดยในระบบ Virtual Memory นี้ หน่วยความจำจะทำหน้าที่คล้ายกับช่องเปิดของเนื้อที่ที่อยู่ในหน่วยความจำสำรองทั้งหมด เทคนิคนี้จะใช้ได้ผลดีถ้าช่องเปิดนี้มีขนาดใหญ่พอ และใช้ในตำแหน่งที่ถูกต้อง

การออกแบบ VIRTUAL MEMORY

หลักการเบื้องต้น

ในระบบ Virtual memory นี้จะต้องอาศัยการควบคุมทั้งด้านฮาร์ดแวร์และด้านซอฟท์แวร์รวมกัน ถ้าหากเราเข้าใจโครงสร้างและการทำงานของ Virtual memory ดี ก็สามารถที่จะนำไอซีประเภท MMU (Memory Management Unit) มาใช้งานได้

ในระบบคอมพิวเตอร์นั้น *Address Space (AS) ก็คือขนาดของหน่วยความจำทั้งหมดซึ่งเราสามารถอ้างถึง ได้ด้วยชุดของคำสั่ง* ซึ่งขนาดของ Address Space นี้ ขึ้นอยู่กับจำนวนบิทของแอดเดรสรีจิสเตอร์ ส่วน *logical address หรือแอดเดรสที่อ้างถึงในโปรแกรม*นั้นจะบอกถึงตำแหน่งของหน่วยความจำที่ CPU จะใช้ส่วน *physical address นั้นจะบอกถึงตำแหน่งของหน่วยความจำที่จะถูกใช้จริงในหน่วยความจำหลัก* แต่เดิมนั้นในระบบไมโครคอมพิวเตอร์ทั่ว ๆ ไป ทั้ง logical address และ physical address ก็คือสิ่งเดียวกันนั่นเอง (อย่างเช่นในหน่วยความจำของไมโครคอมพิวเตอร์ขนาด 8 บิท) แต่ในระบบ Virtual memory นี้ขนาดของแอดเดรสสเปซ (AS) จะเสมือนกับว่าถูกขยายให้ใหญ่กว่าหน่วยความจำหลักที่มีอยู่จริง โดยรวมถึงบริเวณในหน่วยความจำสำรองในดิสค์ด้วย ส่วน logical address ก็ไม่ตรงกับแอดเดรสที่ใช้จริง ๆ ใน RAM และคำสั่งต่าง ๆ สามารถอ้างถึงข้อมูลหรือโปรแกรมใน AS ส่วนที่ยังไม่ได้อยู่ในหน่วยความจำหลักได้ด้วย

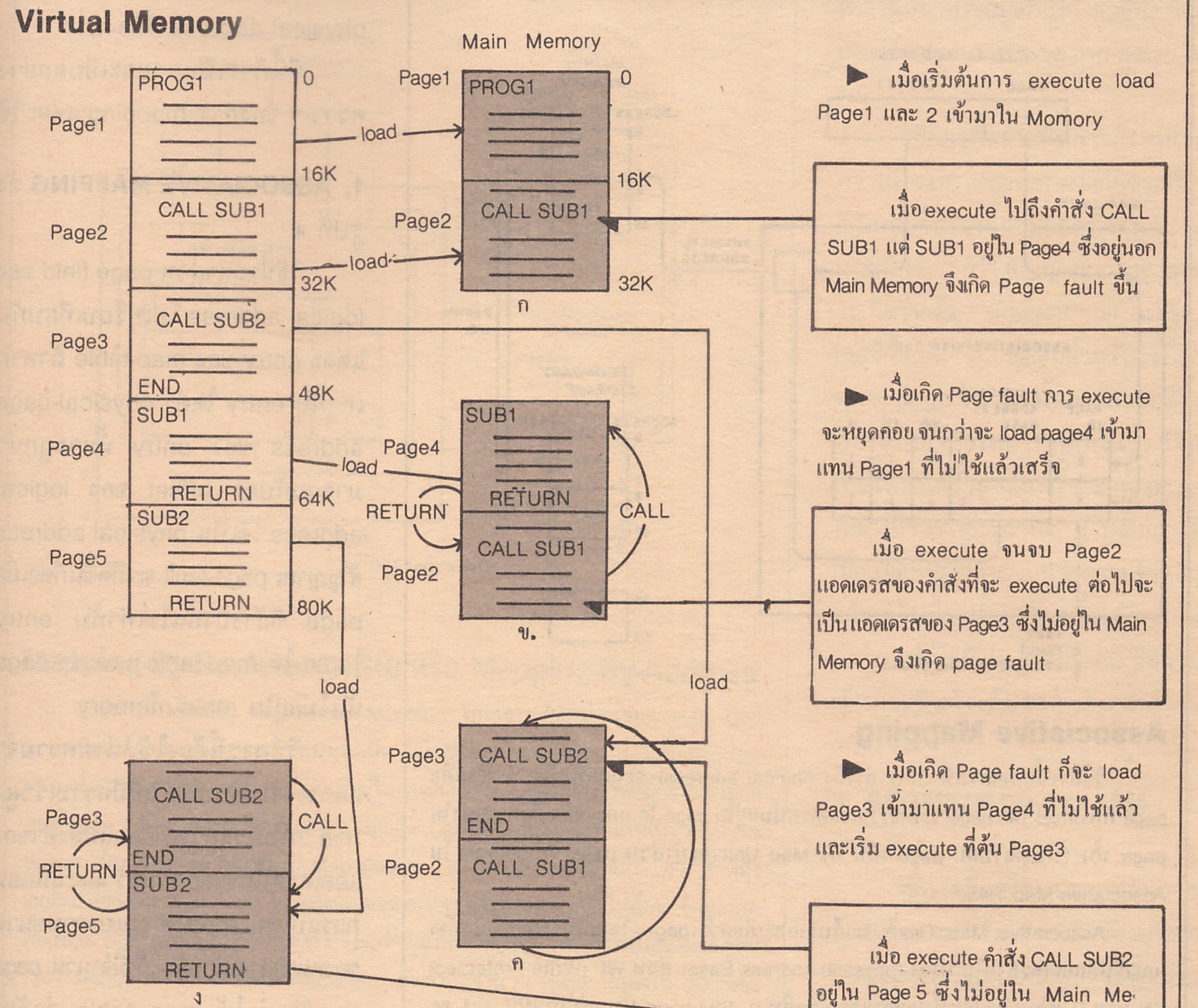
ในระบบ Virtual Memory นั้น AS *ที่อยู่ในดิสค์จะถูกแบ่งเป็นส่วนย่อย ๆ ที่มีขนาดเท่า ๆ กัน เรียกว่า page หรือส่วนย่อย ๆ ที่มีขนาด ไม่เท่ากันเรียกว่า segment เวลา CPU ทำการ execute คำสั่งที่อยู่ใน page ใดก็ต้องนำข้อมูลหรือโปรแกรมที่อยู่ใน page นั้นจากดิสค์เข้าไปไว้ในหน่วยความจำหลัก (RAM) ซึ่งเรียกว่าการนำ page เข้าหรือ page-in และเมื่อไม่ใช้แล้วก็นำ-page นั้น (ซึ่งอาจมีข้อมูลเปลี่ยนแปลงไปเนื่องจากการ execute)ออกไปเก็บในดิสค์ตามเดิม เรียกว่า page-out การนำ-page เข้า-ออกจากหน่วยความจำหลักหรือที่เรียกว่าการทำ paging นี้จะต้องทำให้น้อยที่สุด เพื่อไม่ให้เสียเวลาในการ-execute มากนักเพราะดิสค์มีอัตราการรับ-ส่งข้อมูลช้ากว่าหน่วยความจำหลักมากดังที่กล่าวแล้ว*

ในระหว่างการ execute ถ้า CPU อ้างถึง logical address ใน AS ส่วนที่ยังไม่ได้อยู่ใน RAM จะเกิด error ที่เรียกว่า page fault ขึ้น ระบบ Virtual Memory จะต้องตรวจสอบว่ามี page fault เกิดขึ้นหรือไม่ (page-fault detection) และต้องสามารถย้ายข้อมูลในแอดเดรสสเปซส่วนที่ต้องการนั้นลงมาในหน่วยความจำหลักได้ และเมื่อส่วนที่ย้ายเข้ามาใหม่นั้นถูกชุดคำสั่งอ้างถึงแล้วระบบก็จะต้องสามารถเปลี่ยน logical address ที่ถูกอ้างถึงให้เป็น physical address ที่สมนัยกันเพื่อที่จะ ได้อ้างถึงข้อมูลหรือโปรแกรมส่วนที่ต้องการใช้จริง ๆ ในหน่วยความจำหลักได้ กระบวนการนี้เรียกว่า mapping ทั้งการทำ page-fault detection และ mapping นั้นจะต้องจัดการด้วยฮาร์ดแวร์ที่มีความเร็วสูงมาก

การเกิด page fault จะหยุดยั้งการทำงานของ CPU ซึ่งอาจจะกำลังทำการ Fetch คำสั่งหรือประมวลข้อมูลไว้จนกว่าข้อมูลในหน่วยความจำส่วนที่ต้องการจะได้เข้ามาอยู่ในหน่วยความจำหลักแล้ว

การเกิด page fault นี้จะคล้ายกับการอินเทอร์รัพ (interrupt) แต่ต่างกันตรงที่ page fault อาจจะเกิดขึ้นเมื่อใดก็ได้ (คือเกิดผลให้ CPU หยุดทำงานทันทีในระหว่างที่กำลัง execute คำสั่งอยู่ก็ได้) ส่วนการเกิดอินเตอร์รัพท์จะเกิดขึ้นระหว่างกำลัง execute อยู่ได้ก็จริง แต่CPU จะ execute คำสั่งที่ค้างอยู่จนเสร็จก่อนค่อยมาสนใจอินเตอร์รัพท์ที่เกิดขึ้น

วิธีการ mapping สำหรับ Virtual Memory นั้นมีได้หลายแบบ ดังในรูปที่ 1 ถึงรูปที่ 3 นั้นเป็นวิธี mapping ที่ใช้กันอย่างแพร่หลาย logical address จะประกอบด้วย 2 ส่วนคือ page field และ offset field ส่วน page field นี้จะถูกเปลี่ยนให้เป็น physical address โดย map unit แล้วบวกแอดเดรสด้วย



การจัด Memory ในระบบ Virtual จะแบ่งโปรแกรมและ data ใน Address Space (ซึ่งอยู่ในดิสค์) ออกเป็น page ขนาดเท่า ๆ กัน (หรือถ้าขนาดไม่เท่ากันก็เรียกว่า segment) ในตัวอย่างสมมติให้โปรแกรมใน Address Space มีขนาด 80K แบ่งออกเป็น 5 page (page ละ 16K) ส่วน Main Memory (RAM) ที่มีอยู่จริงมีขนาด 32K ดังนั้นจึงมี page อยู่ใน Main Memory ได้พร้อมกัน 2 page

ในระหว่างการ execute จะมีการ load page ที่จำเป็นต้องใช้เข้ามาใน memory (เรียกว่าการ page-in) และเมื่อไม่ใช้แล้วหากข้อมูลใน page นั้นถูกแก้ไข (อย่าลืมว่าใน page นั้นจริง ๆ แล้วอาจมีทั้งส่วนที่เก็บโปรแกรมและส่วนที่เก็บ data เช่นค่าของตัวแปรต่าง ๆ) ก็จะต้องนำ page นั้นไปเก็บในดิสค์ด้วย (เรียกว่าการ page-out) การ page-in และ page-out จะต้องทำด้วยความเร็วสูง การตรวจสอบว่าเกิด page fault หรือไม่ และการแปลงแอดเดรสที่อ้างถึงในโปรแกรม (แอดเดรสตามที่จัดอยู่ใน Address Space ในดิสค์) กับแอดเดรสจริงใน RAM (ซึ่งขึ้นอยู่กับว่า page นั้นจะถูก page-in ไปอยู่ในส่วนใดของ RAM) คือกระบวนการที่เรียกว่า Dynamic Address Translation หรือ mapping ซึ่งจะถูกกระทำโดยฮาร์ดแวร์ความเร็วสูงที่สร้างขึ้นเพื่อทำหน้าที่นี้โดยเฉพาะ

รูปที่ 3

ค่าที่ได้มาจาก offset field ก็จะได้ physical address ที่ต้องการ

ทีนี้ก็จะเป็นรายละเอียดอย่างคร่าว ๆ ของการ mapping แต่ละวิธี

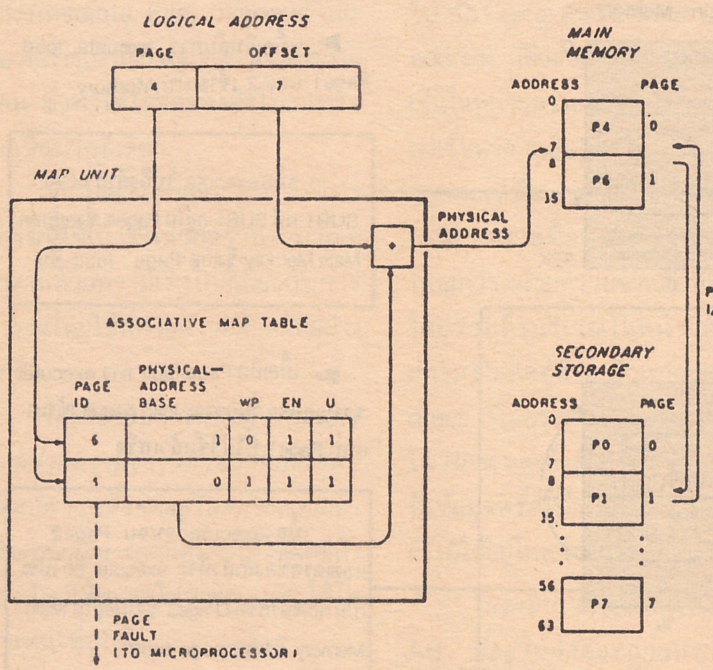
1. ASSOCIATIVE MAPPING ดังรูปที่ 4

วิธีนี้จะนำส่วน page field ของ logical address ไปเปรียบเทียบกับแต่ละ entry ของ map-table ถ้าหากเท่ากับ entry ใดค่า physical-page address ของ entry นั้นจะถูกนำมารวมกับค่า offset ของ logical address ได้เป็น physical address สัญญาณ page-fault จะเกิดขึ้นก็ต่อเมื่อ page ที่อ้างถึงนั้นไม่เท่ากับ entry ใดเลย ใน map table แสดงว่า page นั้นไม่อยู่ใน main memory

วิธีการนี้ต้องใช้หน่วยความจำและวงจรเปรียบเทียบที่มีความเร็วสูงมาก กระบวนการเปลี่ยนแปลงค่าของแอดเดรสนั้นจะต้องรวดเร็วและแน่นอน การแบ่งขนาด page จะแบ่งตามขนาดของหน่วยความจำหลัก ถ้ามีจำนวน page มากก็จะทำให้ map table จำต้องมีขนาดใหญ่ด้วย และถ้าคุณเปลี่ยนขนาดของหน่วยความจำหลักคุณก็ต้องเปลี่ยนขนาดของ map table ด้วย วิธีนี้จะเหมาะสมสำหรับระบบที่มีการแบ่ง page ในแอดเดรสสเปซมาก ๆ และหน่วยความจำหลักที่มีขนาดคงที่ซึ่งไม่ใหญ่นัก

2. MAPPING BY REGISTER (MBR) หรือ MAPPING BY ADDRESS ดังรูปที่ 5

ในวิธีการนี้ ส่วน page field ของ logical address จะไปอ้างถึงชุดของรีจิสเตอร์ที่มีความเร็วสูง ซึ่งในรีจิสเตอร์



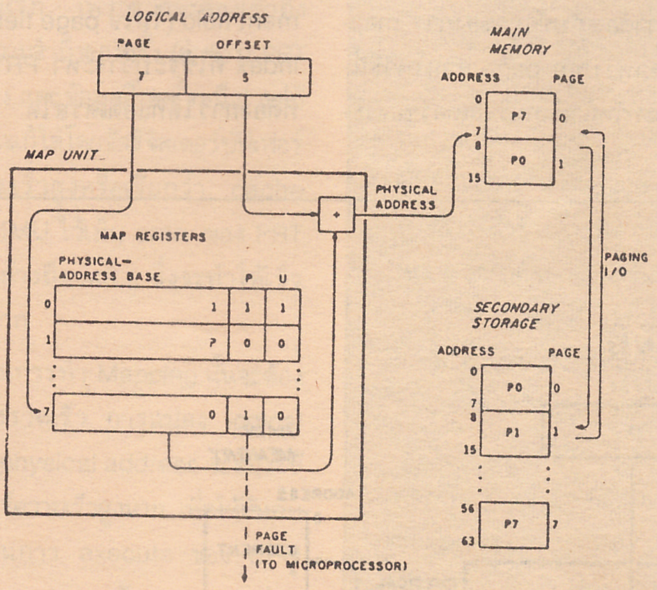
Associative Mapping

วิธีการนี้แอดเดรสที่ CPU อ้างถึง (logical address) จะถูกแบ่งเป็น 2 ส่วนคือ page กับ offset โดย page จะบอกว่าแอดเดรสนั้นอยู่ใน page ใด และ offet คือตำแหน่งใน page นั้น (นับจุดเริ่มต้น page เป็น 0) Map Unit จะนำส่วน page ไป search ใน Associative Map Table

Associative Map Table จะเก็บรายละเอียดว่า page ใดอยู่ใน Memory บ้าง และเริ่มที่แอดเดรสใดใน RAM (physical-address Base) ส่วน WP (Write Protected) EN และ U เป็น flag bit ที่บอกรายละเอียดอื่น ๆ ของ page นั้น เช่นถ้า WP = 1 จะแก้ไขข้อมูลใน page นั้นไม่ได้ ถ้า EN เป็น 1 ทุก page แสดงว่า Main Memory (RAM) เต็มแล้ว ไม่มีที่ให้ load page ใหม่ (นอกจากจะ page-out ก่อน หรือ load ไปทับบาง page ที่ไม่ใช้แล้ว) ส่วน U บอกว่า page นั้นเพิ่งถูกใช้เมื่อไม่นานมานี้ (ถูกใช้บ่อย) ที่จริงจะมีรายละเอียดอื่น ๆ มากกว่านี้อีกก็ได้

เมื่อ Map Unit ทำการ search ใน Associative Map Table ถ้าพบแสดงว่า page นั้นถูก load เข้ามาอยู่แล้ว จึงไม่เกิดสัญญาณ page fault ที่ Map Unit จะต้องทำก็เพียงแต่แปลง (translate) แอดเดรสจาก logical address ให้เป็น physical address เท่านั้น โดยบวกแอดเดรสตั้งต้นของ page (physical-address base) จาก Table เข้ากับ offset เช่นในตัวอย่าง page 4 ซึ่งถูกอ้างถึงอยู่ใน Memory แล้ว จึงมีข้อมูลเก็บไว้ใน Table เมื่อ Map Unit ทำการ search ก็พบและทำการแปลงแอดเดรส

ถ้าทำการ search ไม่พบแสดงว่า page นั้นไม่อยู่ใน Main Memory ก็จะมีสัญญาณ page fault ไปยัง CPU เพื่อให้ไปสั่งอุปกรณ์ในการทำ paging (Paging I/O) ให้ทำการ page-in หรือ page-out ระหว่าง Main Memory (RAM) และ Secondary Storage (Disk) ตามความจำเป็นต่อไป



Mapping-By-Register (MBR) หรือ Mapping-By-Address

วิธีการนี้คล้ายกับ Associative Mapping นอกจากว่าในส่วนของ Map Unit จะประกอบด้วย Map register แทนที่จะเป็น Associative Map Table ใน Map Register จะไม่เก็บเฉพาะรายละเอียดของ page ที่อยู่ใน Main Memory เท่านั้น แต่จะเก็บรายละเอียดของ page ทุก ๆ page ใน Address Space (ในดิสค์) ทีเดียว เช่นถ้าแบ่ง Address Space เป็น N page ก็จะต้องมี Map Register N ตัวที่คู่กัน ดังนั้นเมื่อ CPU อ้างถึง logical address ส่วน page ใน logical address จะทำให้ Map Unit ไปค้นหาข้อมูลจาก Register ที่คู่กับ page นั้น ๆ โดยตรง จึงหาข้อมูลเกี่ยวกับ page นั้นได้ทันทีโดยไม่ต้องเสียเวลา search หา แต่จะต้องมี Register มากเท่ากับจำนวน page ที่แบ่งใน Address Space

ใน Map Register จะเก็บแอดเดรสตั้งต้นของ page นั้นใน Main Memory (ถ้า page นั้นอยู่ใน RAM แล้ว) และมี flag bit P หมายถึง "present" คือถ้าเป็น "1" ก็แสดงว่า page นั้นอยู่ใน RAM ถ้าเป็น "0" ก็ยังไม่อยู่ เมื่อ Map Unit หาข้อมูลของ page ใด ๆ จาก Register ถ้า P = 1 ก็จะทำเพียงแต่การบวก offset กับแอดเดรสเริ่มต้นของ page ใน RAM เพื่อให้ได้ physical address แต่ถ้า P = 0 แสดงว่า page นั้นยังไม่อยู่ใน RAM ก็จะเกิดสัญญาณ page fault ทำให้อุปกรณ์ Paging I/O ทำการ page-in, page-out ระหว่าง RAM กับ Disk ตามความจำเป็นต่อไป

ดังในภาพ page 7 ถูกอ้างถึงใน logical address ข้อมูลใน Map Register ของ page 7 มี P = 1 แสดงว่า page 7 อยู่ใน RAM แล้ว จาก Map Register บอกว่าอยู่ที่ page 0 ใน RAM (คือมีแอดเดรสตั้งต้น page = 0) ดังนั้นค่า offset จึงถูกบวกกับ 0 เป็น physical address แต่ถ้าเกิดใน logical address อ้างถึง page 1 แทน Map Register ของ page 1 มี P = 0 จึงต้อง page-in เข้ามา ซึ่งน่าจะเข้ามาแทนที่ page 7 ก็ได้เพราะ page 7 มี U = 0 แสดงว่า 0 แสดงว่าไม่ได้ถูกใช้บ่อยนัก (U จะเป็นตัวช่วยในการวิเคราะห์ว่าควรเอา page ใดเข้ามาแทน page ใด)

เหล่านี้จะเก็บข้อมูลที่เป็น status และ physical address ของทุก page ใน AS.

วิธีการนี้คล้ายกับการอ้างหน่วยความจำแบบ indirect ต่างกันตรงที่ว่ารีจิสเตอร์เหล่านี้สามารถที่จะถูกประมวลผลได้เร็วกว่า สัญญาณ page fault จะเกิดขึ้นเมื่อ "present" bit ในรีจิสเตอร์นั้นเป็น "0" ข้อเสียสำหรับวิธีนี้คือทุก ๆ page ใน AS นั้นจะต้องถูกอ้างถึงได้โดยรีจิสเตอร์ที่คู่กันในชุดรีจิสเตอร์ความเร็วสูงนั้น ถ้ามีมาก page ก็ต้องมีรีจิสเตอร์มากตามไปด้วย แต่เมื่อเปลี่ยนขนาดของหน่วยความจำหลักแล้วจะไม่มีผลต่อการ map ด้วยส่วนฮาร์ดแวร์เช่นนี้เลย และถ้า AS มีขนาดเล็กก็จะยิ่งทำให้ประสิทธิภาพการ-map ดีมากขึ้น

3. SEGMENTED-SYSTEM

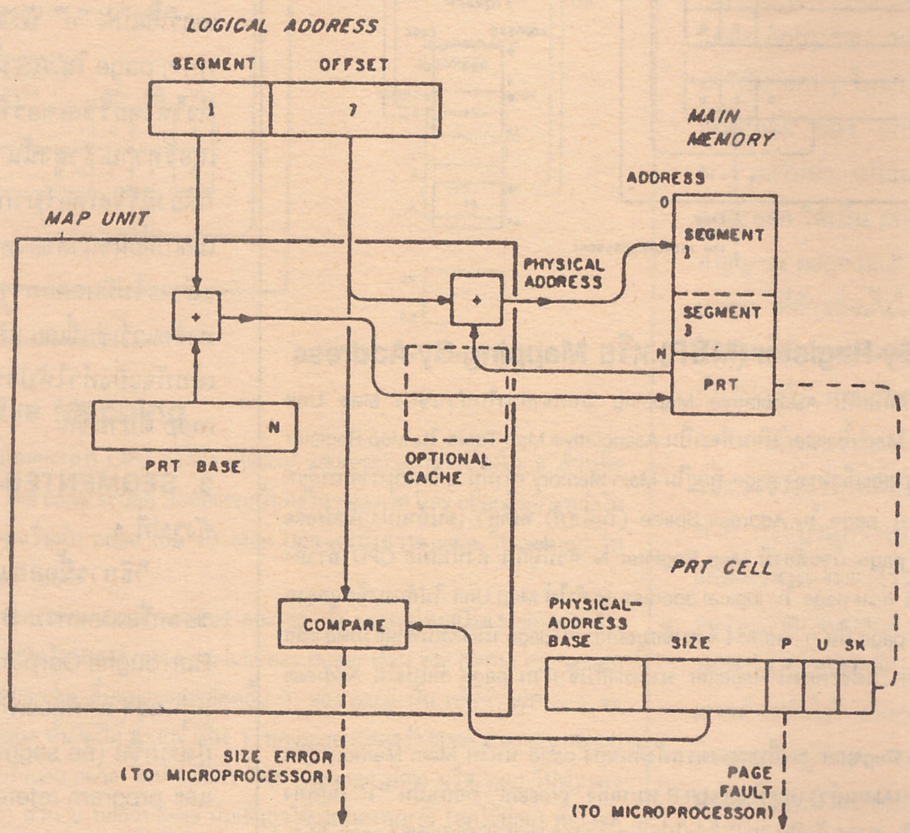
ดังรูปที่ 6

วิธีการนี้ออกแบบจากหลักการของเครื่องเมนเฟรม B 5500 ของบริษัท Burroughs Corporation วิธีนี้จะยืดหยุ่นกว่า แต่ก็ต้องเพิ่มเติมฮาร์ดแวร์ที่ยุ่งยากขึ้น (คือ segmented-size field และ program reference table หรือ-PRT) ส่วน segment field จะมาแทนส่วน page field ใน logical address ในระบบของคุณจะมีกี่ segment ก็ได้ ขนาดของหน่วยความจำหลักและแอดเดรสสเปซจะไม่มีผลต่อการ map

การเปลี่ยนแอดเดรสนั้นเหมือน ๆ กับ mapping by address ส่วน program reference table (PRT) นั้นจะเก็บขนาดและตำแหน่งของ segment นั้น ๆ จะ PRT นี้จะถูกเก็บในหน่วยความจำหลักแทนที่จะใช้รีจิสเตอร์ภายใน Map Unit ทั้งนี้เพราะ

1) หากว่าจำนวน segment มีมากหรือเปลี่ยนแปลง จะได้ไม่ต้องเปลี่ยนฮาร์ดแวร์ใน Map Unit

2) หลีกเลี่ยงการใช้รีจิสเตอร์ ซึ่งถ้าใช้ก็จะต้องมีเป็นจำนวนมาก เพราะว่าต้องเก็บข้อมูลของการ mapping สำหรับทุก page ส่วนในวิธีนี้-PRT จะทำหน้าที่แทนรีจิสเตอร์ของวิธี-Mapping-by-Register และมี segment field (ไม่ใช่ page field) เป็น-index ที่ใช้ระบุว่าจะหา PRT CELL ที่ต้องการได้ที่แอดเดรสใด



Segmented Mapping System

วิธีนี้เก็บข้อมูลของทุก ๆ page ใน Address Space เช่นเดียวกับ Mapping-By-Register แต่แทนที่จะเก็บในกลุ่มของ Register ใน Map Unit ก็เก็บในส่วนหนึ่งของ Main Memory แทน เนื้อที่ส่วนนี้เรียกว่า Program Reference Table (PRT) โดยใน Map Unit จะมีเพียง Register ตัวเดียวที่เก็บแอดเดรสตั้งต้นของ PRT ใน Main Memory เท่านั้น คือ PRT BASE

นอกจากนั้นวิธีการนี้ยังแบ่ง Address Space ออกเป็นหลาย segment ซึ่งมีขนาดไม่เท่ากัน แทนที่จะแบ่งเป็น page เท่า ๆ กันอย่างวิธีอื่น ดังนั้นใน PRT จะต้องเก็บข้อมูลส่วนที่บอกขนาดของแต่ละ segment ไว้ด้วยโดย PRT จะแบ่งออกเป็นหลาย CELL แต่ละ CELL ก็สำหรับ segment หนึ่ง ๆ และแต่ละ CELL ก็จะมีข้อมูลทั้งขนาดและแอดเดรสตั้งต้นของ segment, present bit (P), Used bit (U) และยังอาจมีข้อมูลอื่น ๆ ที่บอกลักษณะพิเศษของข้อมูลใน segment อีกด้วย เช่น SK บอกว่าเป็น stack segment หรือไม่ (ถ้าเป็นการใช้ค่า offset ใน segment นั้นจะกลับกับปกติ)

ในการ Mapping แบบนี้ logical address จะมี 2 ส่วนคือ segment (แทน page) และ offset แต่วิธีการ Mapping จะยุ่งขึ้นเล็กน้อยโดยเมื่อ CPU ส่ง logical address มา ส่วน segment จะถูกนำไปบวกกับแอดเดรสตั้งต้นของ PRT ใน PRT BASE เช่นถ้า segment = 0 ก็จะหมายถึง CELL ที่ 1 คือ CELL แรกใน PRT ถ้าเป็น N ก็จะหมายถึง CELL ที่ N + 1 ใน PRT เมื่อหา PRT CELL ของ segment นั้น ๆ พบแล้ว เนื่องจากขนาดของแต่ละ segment ไม่เท่ากัน สิ่งแรกที่ต้องทำก็คือตรวจสอบว่าค่า offset ที่กำหนดใน logical address นั้นเกินขนาดของ segment หรือไม่ถ้า COMPARE แล้วเกินก็จะเกิดสัญญาณ SIZE ERROR แจ้งไปยัง CPU ต่อไป

เมื่อตรวจสอบ SIZE แล้วก็ต้องดูว่า segment นั้นอยู่ใน Main Memory หรือยัง ถ้าอยู่ก็แปลง logical ให้เป็น physical address เลย ถ้าไม่อยู่ก็ต้องทำ page-in เข้ามา

เนื่องจาก Main Memory มีอัตราเร็วในการทำงานช้ากว่า Register หรือ Map table ในวิธีอื่น การเข้าถึงข้อมูลใน PRT จึงอาจช้ากว่า แต่ก็สามารถแก้ไขได้โดยใช้ chache memory (Memory ความเร็วสูงขนาดเล็ก) ติดตั้งไว้ใน Map Unit แล้วดึงข้อมูลใน PRT CELL ที่ถูกอ้างถึงบ่อย ๆ เข้าไปเก็บใน chache memory เลย เพื่อลดเวลาในการเข้าถึงข้อมูลใน PRT CELL ลง

จะเห็นได้ว่าวิธีการนี้ช้ากว่าสองวิธีการแรก เพราะว่าจะต้องเข้าถึงหน่วยความจำถึงสองครั้ง (สำหรับ PRT ครั้งหนึ่ง และข้อมูลอีกครั้งหนึ่ง) แต่ก็สามารถแก้ได้โดยจัดรีจิสเตอร์บัฟเฟอร์ที่มีความเร็วสูงหรือเรียกว่า cache ไว้ใน MMU ไว้เก็บ entry ของ PRT ที่มีการอ้างถึงบ่อย ๆ ซึ่งจะทำให้เร็วขึ้นอีกมาก

ในการทำ Mapping นั้นจะต้องทำอย่างรวดเร็ว การแปลง logical ให้เป็น physical address จะต้องใช้ฮาร์ดแวร์ความเร็วสูงมาก เพื่อให้ทันกับจังหวะในการ execute ของ CPU การย้าย page เข้า-ออกจากหน่วยความจำก็จะต้องทำให้น้อยที่สุดเพราะ CPU จะต้องหยุดคอย เนื่องจากความเร็วในการย้ายข้อมูลของดิสค์จะช้ากว่าRAM มากดังที่กล่าวแล้ว

การที่ CPU กับดิสค์จะย้ายข้อมูลให้น้อยที่สุดนั้นจะต้องอาศัยการวิเคราะห์การใช้งานของ page ต่าง ๆ ว่า page ใดใช้บ่อยหรือมากน้อยอย่างไร หลักการก็คือจะต้องพยายามให้ page ที่ใช้บ่อย ๆ อยู่ใน main memory ให้มากที่สุดเพื่อลดโอกาสที่จะต้องย้ายมันเข้ามาบ่อย ๆ เมื่อถูกเรียกใช้ ส่วน page ที่ไม่ค่อยได้ใช้ก็เก็บไว้ในดิสค์ให้มากเพื่อจะได้ไม่มาแย่งที่ของ page ที่ใช้บ่อย ๆ แต่การที่ CPU จะวิเคราะห์สิ่งเหล่านี้ได้ก็จะต้องอาศัยข้อมูลของแต่ละ page ที่เก็บไว้ใน map table หรือ map register เช่น U (Used), WP (Write-Protected), P(Present) และอื่นๆอีกมากแล้วแต่กรณี(ดูรูปประกอบ) ดังนั้นใน map table หรือ map register ก็อาจมีข้อมูลเพิ่มเติมเพื่อบอกลักษณะต่าง ๆ ของ page นั้น ๆ ได้อีกหลาย ๆ

อนาคตของ Virtual Memory กับไมโครคอมพิวเตอร์

ความจริงแล้วแนวความคิดของ Virtual Memory หรือ Virtual Storage นั้นมีใช้มานานแล้วในระดับของเครื่องใหญ่ หรือ เมนเฟรม และยังเป็นระบบที่สนับสนุนการพัฒนาทางซอฟท์แวร์ต่าง ๆ เช่นการจัด File ในระบบ VSAM (Virtual Storage Access Method) แต่เพิ่งจะเร็ว ๆ นี้เองที่เทคโนโลยีทางฮาร์ดแวร์จะก้าวหน้าพอที่จะนำระบบนี้มาใช้กับไมโครคอมพิวเตอร์ได้ ทั้งนี้เพราะข้อจำกัดต่าง ๆ เช่น

- *ไมโครโปรเซสเซอร์มีขีดความสามารถไม่สูงนัก มีความเร็วต่ำ ซอฟท์แวร์ที่ใช้ก็มีขนาดไม่ใหญ่โตเกินขีดความสามารถในการอ้างแอดเดรส (physical address space) ของมัน*
- *อุปกรณ์เก็บข้อมูลภายนอกหรือ secondary storage ของไมโครคอมพิวเตอร์ยังใช้ floppy disk ซึ่งมีความจุน้อย และความเร็วในการย้ายข้อมูลเข้าออกกับ Memory (RAM) ก็ไม่สูงพอที่จะทำให้ระบบ Virtual Memory ทำงานได้อย่างมีประสิทธิภาพ*
- *วงจรทางฮาร์ดแวร์ที่ทำหน้าที่ mapping ยังไม่สามารถทำออกมาให้มีประสิทธิภาพดี ความเร็วสูงและราคาต่ำได้*

ส่วนเครื่องเมนเฟรมนั้นไม่ติดปัญหาเหล่านี้เนื่องจากมีความเร็วสูงและใช้ระบบดิสค์ที่มีความจุมากและความเร็วสูง รวมทั้งมีความเชื่อถือได้ (Reliability) ดีพอเพราะระบบนี้จะต้องมีการย้ายข้อมูลระหว่าง disk และ Memory ในปริมาณสูงมาก แต่ในระยะหลังนี้ไมโครโปรเซสเซอร์ได้รับการพัฒนาให้มีความเร็วสูงขึ้น และมีประสิทธิภาพมากขึ้น ซอฟท์แวร์ที่ใช้ก็ได้รับการพัฒนาตามไปด้วย และมีขนาดใหญ่เพิ่มขึ้นมาก นอกจากนี้ยังมีการนำ Hard disk เข้ามาใช้กับไมโครคอมพิวเตอร์ ทำให้ความเร็วในการย้ายข้อมูลและความเชื่อถือได้สูงขึ้น ดังนั้นจึงมีผู้พัฒนาไอซี Dynamic Address Translation ที่ทำหน้าที่ Mapping สำหรับไมโครคอมพิวเตอร์ขึ้น

แต่ถึงแม้เทคโนโลยีทางฮาร์ดแวร์จะก้าวหน้าไปมากก็ตาม ขีดจำกัดอีกอันหนึ่งของ Virtual Memory ก็คือด้านซอฟท์แวร์เพราะถึงแม้ระบบดิสค์ของทั้งไมโครและเมนเฟรมจะได้รับการพัฒนาให้มีความเร็วสูงขึ้น แต่หน่วยความจำหลักคือ RAM ก็มีความเร็วสูงขึ้นด้วยเช่นกัน ซึ่งเมื่อเปรียบเทียบกันแล้ว การเข้าถึงข้อมูลใน RAM ก็ยังเร็วกว่าในดิสค์มาก ดังนั้นเพื่อให้ระบบ Virtual Memory มีประสิทธิภาพสูงสุดก็จะต้องทำการย้ายข้อมูลระหว่าง RAM กับดิสค์ให้น้อยที่สุด แต่จะทำได้อย่างไรยังเป็นเรื่องที่ต้องพัฒนากันต่อไปในด้านซอฟท์แวร์โปรแกรมควบคุมระบบที่จะใช้ Virtual Memory ได้อย่างมีประสิทธิภาพจะต้องรู้ว่าจะจัดการย้ายข้อมูลอย่างไรจึงจะมีประสิทธิภาพ เมื่อไหร่ควรทำ page-in และ/หรือ page-out กับ page ไหน ฯลฯ

การใช้ Virtual Memory อย่างมีประสิทธิภาพนั้นมักจะพิจารณาใน 2 ประเด็นคือ

ขนาดของ page หรือ segment ควรเป็นเท่าไหร่จึงจะดีที่สุด

- ควรทำ page-in, page-out เมื่อไหร่

สำหรับขนาดของ page หรือ segment นั้นขึ้นอยู่กับปัจจัยต่าง ๆ มากมาย เช่น ลักษณะของงานที่ใช้ ขนาดของโปรแกรมที่ใช้มากขนาดของ module หรือโปรแกรมย่อย ๆ ในแต่ละโปรแกรม ฯลฯ ตัวอย่างเช่นในเครื่องเมนเฟรมของ IBM ที่มี Memory ขนาดหลาย ๆ Megabyte โปรแกรมควบคุมระบบ VM/370 และ VM/SP ใช้การแบ่ง page ขนาด 2K ส่วน VM/SP HPO มีทั้ง 2K และ 4K เป็นต้น

ส่วนที่ว่าเมื่อไหร่ควรทำ page-in, page-out นั้น วิธีการง่ายที่สุดคือ *ทำเมื่อจำเป็นหรือมีความต้องการจะใช้ page นั้น ๆ แล้ว page นั้นไม่อยู่ใน Main Memory* (วิธีนี้เรียกว่า Demand Paging) แต่มีวิธีการที่จะทำให้ประสิทธิภาพของระบบดีขึ้นอีกก็โดย*การอาศัยการวิเคราะห์ page (page analysis)เข้าช่วย* เช่นวิเคราะห์ว่า page ใดใช้บ่อยก็พยายามจัดไว้ใน Memory เลยไม่ page-out ออกมา page ใดไม่ค่อยได้ใช้ก็ page-out ออกมาเพื่อให้มีที่สำหรับ page อื่น ๆ การทำ page analysis นี้ไม่เพียงแต่ระหว่างการ page-in, page-out เท่านั้น แม้แต่บนดิสค์ก็ยังมีการวิเคราะห์ว่า page ใดที่จะใช้มากจะถูกย้ายให้มาอยู่บริเวณที่เข้าถึงได้ง่ายบนดิสค์ ส่วน page ที่ไม่ค่อยได้ใช้ก็จะถูกย้ายไปอยู่ที่ที่เข้าถึงยากเช่นตอนริม ๆ แทน (เรียกว่าการทำ Migration)อีกด้วย วิธีการเหล่านี้จะทำให้ประสิทธิภาพของ Virtual Memory สูงขึ้นอีก

ในปัจจุบันไมโครโปรเซสเซอร์รุ่นใหม่ ๆ ได้พัฒนาไปในทางที่จะใช้ประโยชน์จาก Virtual Memory ให้มากขึ้นเช่น 8086/8088 ของอินเทลมีการแบ่ง Memory เป็น Segment ขนาดไม่เกิน 64K จำนวน 16 segment ครอบคลุม Address Space ทั้งหมด 1M (ดูเรื่อง 8086/8088 ในไมโครฯ ฉบับที่ 8) หรือ 80386 ซึ่งเป็นไมโครโปรเซสเซอร์ 32 บิทของอินเทลที่กำลังจะออกมาก็จะมีกลไกในการทำ Virtual Memory อยู่ภายในเลยเป็นต้น ส่วนไมโครคอมพิวเตอร์ที่สามารถทำ Virtual Memory ได้ที่ออกจำหน่ายแล้วก็คือ XT/370 ของ IBM ซึ่งเป็นรุ่นที่ดัดแปลงจาก IBM XT (XT ต่างจาก PC ธรรมดาที่มี Hard disk ส่วน XT/370 ดัดแปลงจาก XT อีกทีหนึ่ง) โดยเพิ่มแผงวงจรที่ใช้ไมโครโปรเซสเซอร์ 68000 ของโมโตโรลา 2 ตัว ทำหน้าที่ Simulate การทำงานของ CPU 32 บิทของเมนเฟรม IBM 370 (ดูรายละเอียด XT/370 ในข่าวฉบับ8) โดยมีโปรแกรมควบคุมระบบ VM/PC ที่พัฒนาขึ้นเพื่อให้ใช้ Virtual Memory และซอฟท์แวร์ของเมนเฟรม 370 ได้

ระบบ Virtual Memory ในไมโครคอมพิวเตอร์เพิ่งจะเริ่มต้น เทคโนโลยีที่ใช้กันอยู่ในขณะนี้ก็ล้วนแต่เป็นสิ่งที่รับมาจากเมนเฟรมเป็นส่วนมาก แต่ในอนาคตอันใกล้นี้เมื่อไมโครฯ มีระบบ Virtual Memory ใช้กันแพร่หลาย ด้วยความรวดเร็วในการพัฒนาของไมโครฯก็อาจเป็นไปได้ที่ว่าจะมีอะไรใหม่ ๆ ถูกพัฒนาขึ้นบนไมโครฯโดยตรงเลยก็ได้

(วศิน เพิ่มทรัพย์)

อย่างตามความจำเป็น ตัวอย่างของข้อมูลเหล่านั้นดูได้จากตารางท้ายเรื่อง

สรุป

Virtual Memory เป็นระบบการจัด **Memory** ที่เริ่มเข้ามามีบทบาทในไมโครคอมพิวเตอร์เมื่อเร็ว ๆ นี้โดยเป็นแนวความคิดในการใช้หน่วยเก็บข้อมูลภายนอก เช่น ดิสค์มาช่วยขยายขนาด **Address Space** ของ **main memory** ให้เพิ่มขึ้นอีกมาก ทำให้การพัฒนาซอฟท์แวร์สะดวกขึ้นและไม่ถูกจำกัดโดยขนาดของ **main memory** หรือ **RAM**

การใช้ **Virtual Memory** จะต้องมีการแบ่ง **Address Space** ออกเป็น **Page** หรือ **segment** เก็บไว้ในดิสค์ **page** หรือ **segment** นี้จะถูกนำเข้ามาเมื่อต้องการใช้ และถูกนำออกไปเมื่อไม่ใช้ และจะต้องมีวงจรที่ทำหน้าที่ย้าย **page** ที่ต้องการใช้เข้ามาใน-**main memory** ตลอดจนเปลี่ยน **logical address** ที่ **CPU** อ้างถึงใน **Address Space** ให้เป็น **physical address** ตามที่ **page** หรือ **segment** นั้นถูก **load** เข้ามาจริง ๆ ใน **RAM** กระบวนการนี้เรียกว่า **mapping** ซึ่งทำได้หลายวิธีนอกจากนี้การเพิ่มประสิทธิภาพให้แก่ **Virtual Memory** ยังต้องอาศัยการวิเคราะห์การใช้งานของ **page** หรือ **segment** ต่าง ๆ ประกอบด้วย ซึ่งวิธีการเหล่านี้ตลอดจนอุปกรณ์ทางฮาร์ดแวร์ที่สนับสนุนก็ยังได้รับการพัฒนาอยู่อย่างไม่หยุดยั้ง คาดว่าในอนาคต **Virtual Memory** คงจะแพร่หลายไปในเครื่องไมโครคอมพิวเตอร์แบบต่าง ๆ จนอาจกลายเป็นโครงสร้างที่เป็นมาตรฐานของการจัด **Memory** ในไมโครฯไปก็ได้ ■

เล่น Apple GRAPHIC ใน CP/M

โดย อาร์ทีเมีย

ใน CP/M นั้นเราสามารถเล่น Graphic ได้เช่นเดียวกับที่เล่นใน Applesoft BASIC แต่อาจจะแตกต่างกันบ้างตรงวิธีการ และรายละเอียดบางอย่าง ผู้ที่เคยเล่น Graphic ด้วย GBASIC อาจจะเคยพบปัญหาบางอย่าง เช่น ไม่สามารถพิมพ์ข้อความใน 4 บรรทัดสุดท้าย เมื่อใช้คำสั่ง- GR หรือ HGR แต่ความจริงแล้ว

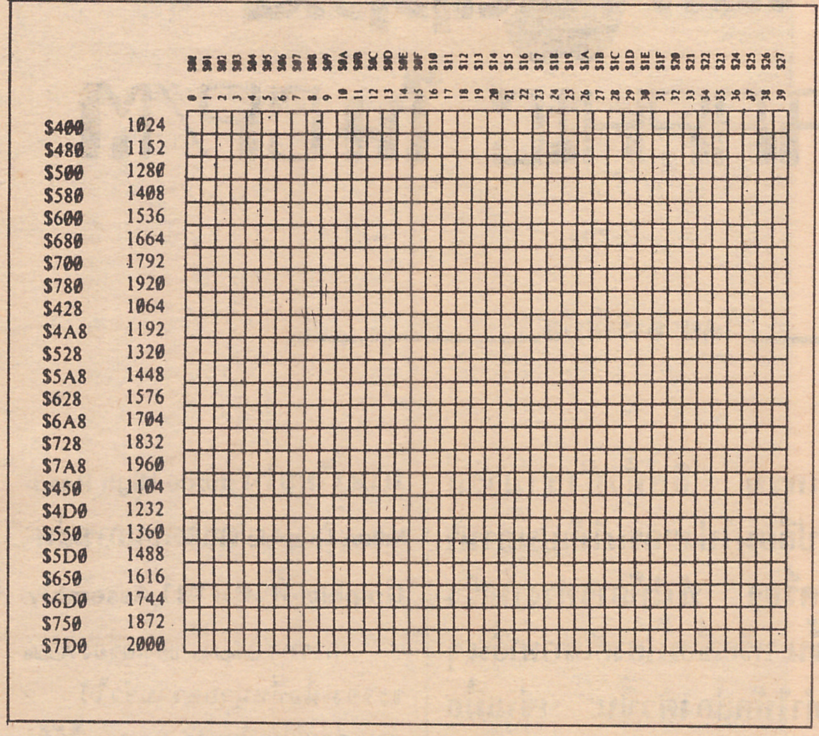
การเล่น Graphic ใน CP/M นี้น่าสนใจและน่าสนุกกว่าการเล่นใน Applesoft BASIC เพราะเราสามารถเล่น- Graphic ได้ด้วยภาษา BASIC, ด้วยภาษา Assembly หรือด้วยภาษาอื่น ๆ **นอกจากนี้ ยังมี Package ต่าง ๆ ที่ช่วยอำนวยความสะดวกในการสร้างโปรแกรม อีกทั้งยังสามารถเปลี่ยนโปรแกรมให้เป็นภาษาเครื่อง ทำให้การทำงานเร็วขึ้น หรือเชื่อมโยงกับภาษาอื่น ๆ ทำให้คล่องตัวขึ้น เช่นเมื่อใดต้องการความเร็วสูงก็เขียนด้วย Assembly เมื่อใดโปรแกรมซับซ้อนก็เขียนด้วยโปรแกรมชั้นสูง ทำให้เราสามารถเขียนโปรแกรมได้ดีขึ้น และที่สำคัญ CP/M เป็น Operating System ที่เป็นมาตรฐานกว่า**

บทความนี้อธิบายการจัดหน่วยความจำสำหรับการทำ Graphic ทั้งสองแบบ การพิมพ์ใน Text 40 Column เมื่อใช้ GBASIC โดยที่มี Card 80 Column เสียบอยู่ และแนวความคิดในการพิมพ์ตัวอักษรภาษาไทยใน mode High Resolution Graphic ตลอดจนการเล่น- Graphic ด้วยภาษา Assembly

การทำ Graphic ของเครื่อง Apple จะจองเนื้อที่หน่วยความจำไว้ 2 ส่วน ส่วนแรกมีขนาด 1 K Byte ไว้สำหรับเก็บ Text หรือ Low Resolution Graphic อีกส่วนมีขนาด 8K Byte สำหรับเก็บ High Resolution Graphic ซึ่งทั้งสองส่วนนี้จะต่ออยู่กับจอภาพโดยตรงโดยขึ้นอยู่กับการสวิทช์ว่าจะให้ส่วนไหนออกจอภาพ การเปลี่ยนแปลงรูปที่จะขึ้นบนจอภาพก็ทำได้โดยเปลี่ยน-แปลงค่าในหน่วยความจำส่วนนี้

GR & TEXT

จอภาพของเครื่อง Apple มีอยู่ 24 แถว แต่ละแถวก็มี 40 Column ดังนั้น ก็จะมีจำนวนตัวอักษรบนจอภาพได้ 960 ตัว ซึ่งแต่ละตัวก็ใช้เนื้อที่ 1 Byte จึงใช้เนื้อที่ในหน่วยความจำเพียง 960 Byte แต่เครื่องจองเนื้อที่ไว้ 1024 Byte หรือ 1K ดังนั้นก็จะมีหน่วยความจำบางส่วนที่สูญเปล่าไป ซึ่งเป็นส่วนที่ไม่ควรจะไปยุ่งเกี่ยวด้วยใน CP/M



รูปที่ 1 address ของ Text screen

หน่วยความจำที่ว่านี้เริ่มที่ address 0400 ถึง 0800

ทดลอง RUN โปรแกรมที่ 1 ซึ่งเป็น Applesoft Basic แล้วคุณจะเข้าใจการจัดหน่วยความจำยิ่งขึ้น

```
10  HOME
20  FOR I = &H0400 TO &H07FF
30  POKE I, ASC ("A")
40  NEXT I
```

โปรแกรมที่ 1 พิมพ์ตัวอักษร A บน Text screen

ใน CP/M address ต่าง ๆ จะถูกเลื่อนไปตามตาราง 1 เช่น ที่ address 0400 นี้ เมื่อต้องอ้างถึงใน CP/M จะต้องอ้างถึงด้วย address F400

Z-80 ADDRESS	APPLE ADDRESS
$0000-$0FFF	$1000-$1FFF
$1000-$1FFF	$2000-$2FFF
$2000-$2FFF	$3000-$3FFF
$3000-$3FFF	$4000-$4FFF
$4000-$4FFF	$5000-$5FFF
$5000-$5FFF	$6000-$6FFF
$6000-$6FFF	$7000-$7FFF
$7000-$7FFF	$8000-$8FFF
$8000-$8FFF	$9000-$9FFF
$9000-$9FFF	$A000-$AFFF
$A000-$AFFF	$B000-$BFFF
$B000-$BFFF	$D000-$DFFF
$C000-$CFFF	$E000-$EFFF
$D000-$DFFF	$F000-$FFFF
$E000-$EFFF	$C000-$CFFF
$F000-$FFFF	$0000-$0FFF

ตารางที่ 1 เปรียบเทียบ address ของ-CP/M กับ Apple

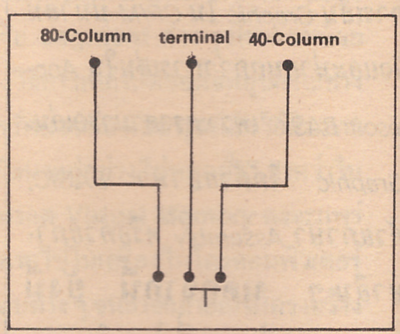
ดังนั้น ถ้าคุณจะทดลอง RUN โปรแกรมที่ 1 ใน MBASIC หรือ GBASIC ซึ่งอยู่ใน CP/M คุณก็ต้องแก้ไข address ในบรรทัด 20

```
10  FOR J = &HF400 TO &HF780
    STEP &H80
20  FOR I = 0 TO & H77
30  POKE I, ASC ("A")
40  NEXT I
50  NEXT J
```

โปรแกรมที่ 2 พิมพ์ตัวอักษร A บน Text screen ใน CP/M

เมื่อ RUN โปรแกรมที่ 2 แล้ว ตัวอักษร A จะไปปรากฏที่ Text screen 40 Column ส่วนที่ 80 Column จะไม่มีอะไรเกิดขึ้น

ใน CP/M ถ้าคุณใช้ 80-Column ด้วย เมื่อคุณพิมพ์อะไรก็ตามมันก็จะมาปรากฏที่ 80-Column แต่การเล่น Graphic ยังคงไปปรากฏที่ 40-Column เช่นเดิม ดังนั้นคุณคงต้องสวิทช์ไปมาระหว่าง 80-Column และ 40-Cloumn ซึ่งวิธีที่จะสวิทช์ก็มีอยู่ด้วยกัน 3 วิธี คือ วิธีแรกใช้มือสลับสายไปมา ซึ่งวิธีนี้ถ้าคุณทำไปซักพักก็คงจะเบื่อ วิธีที่สองใช้ Softswitch ราคาประมาณ 200 บาท ซึ่งเราสามารถใช้ Software ในการสวิทช์ไปมาได้ ส่วนวิธีสุดท้ายเป็นวิธีง่าย ๆ ที่ได้ผลดี วิธีที่ว่า คือ Hardswitch



รูปที่ 2 วงจรสวิทช์ระหว่าง 80 Column กับ Column

ว่าง่าย ๆ ก็คือ ใช้สวิทช์ตัวหนึ่งเป็นตัวสวิทช์ไปมาโดยใช้มือโยกนั่นเอง

สายที่นำมาต่อตามวงจร คือสายสัญญาณซึ่งอยู่ในสุดของสายไพ่ ส่วนสายกราวด์นำมาต่อกันโดยตรงได้เลย

การเล่น Low Resolution Graphic คุณสามารถเล่นได้ใน GBASIC ซึ่งคำสั่งต่าง ๆ เหมือนกับใน Applesoft Basic เช่นคำสั่ง GR VLIN HLIN เป็นต้น ซึ่งจะไม่ขอกล่าวถึงในที่นี้

คราวนี้ก็ถึงปัญหาที่ตั้งไว้ในตอนต้น ตรงที่ว่าเราจะพิมพ์ข้อความใน Text Screen ได้อย่างไร ซึ่งทางออกของเราก็คงจะต้องใช้คำสั่ง POKE แทนคำสั่ง PRINT เช่นในโปรแกรมที่ 3 เป็นการพิมพ์คำว่า APPLE ที่บรรทัดที่ 3 ของ Text screen 40-Column

```
10 A$ = "APPLE"
20 FOR I = 1 TO LEN (A$)
30 POKE &HF500 + I, ASC (MID$
   (A$,I,1))+128
40 NEXT I
```

โปรแกรมที่ 3 พิมพ์คำว่า APPLE ที่ Text screen

ในโปรแกรม 3 นี้ จะแยกตัวอักษรออกมาทีละตัว แล้ว POKE ค่า ASCII ของตัวอักษรนั้นไปที่ address F500+1 ตัวเลข 128 ที่บวกเพิ่มเข้าไปในบรรทัด 30 นั้นเป็นการปรับค่า ASCII ของตัวอักษรให้ตรงกัน

จากตารางที่ 2 และที่ 3 จะเห็นได้ว่า ASCII ของตัวอักษร A ในCP/M ซึ่งมีค่าเท่ากับ 41 จะตรงกับ A ชนิด

Table 7: ASCII Screen Characters

	Inverse				Flashing				Normal							
									(Control)						(Lowercase)	
Decimal	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
Hex	$00	$10	$20	$30	$40	$50	$60	$70	$80	$90	$A0	$B0	$C0	$D0	$E0	$F0
0 $0	@	P		0	@	P		0	@	P		0	@	P		0
1 $1	A	Q	!	1	A	Q	!	1	A	Q	!	1	A	Q	!	1
2 $2	B	R	"	2	B	R	"	2	B	R	"	2	B	R	"	2
3 $3	C	S	#	3	C	S	#	3	C	S	#	3	C	S	#	3
4 $4	D	T	$	4	D	T	$	4	D	T	$	4	D	T	$	4
5 $5	E	U	%	5	E	U	%	5	E	U	%	5	E	U	%	5
6 $6	F	V	&	6	F	V	&	6	F	V	&	6	F	V	&	6
7 $7	G	W	'	7	G	W	'	7	G	W	'	7	G	W	'	7
8 $8	H	X	(	8	H	X	(	8	H	X	(	8	H	X	(	8
9 $9	I	Y	)	9	I	Y	)	9	I	Y	)	9	I	Y	)	9
10 $A	J	Z	•	:	J	Z	•	:	J	Z	•	:	J	Z	•	:
11 $B	K	[	+	;	K	[	+	;	K	[	+	;	K	[	+	;
12 $C	L	\	,	<	L	\	,	<	L	\	,	<	L	\	,	<
13 $D	M	]	–	=	M	]	–	=	M	]	–	=	M	]	–	=
14 $E	N	^	.	>	N	^	.	>	N	^	.	>	N	^	.	>
15 $F	O	_	/	?	O	_	/	?	O	_	/	?	O	_	/	?

ตารางที่ 2 ตารางค่า ASCII ของ Apple

ASCII Code	Character	ASCII Code	Character	ASCII Code	Character
000	NUL	043	+	086	V
001	SOH	044	,	087	W
002	STX	045	-	088	X
003	ETX	046	.	089	Y
004	EOT	047	/	090	Z
005	ENQ	048	0	091	[
006	ACK	049	1	092	\
007	BEL	050	2	093	]
008	BS	051	3	094	↑
009	HT	052	4	095	<
010	LF	053	5	096	'
011	VT	054	6	097	a
012	FF	055	7	098	b
013	CR	056	8	099	c
014	SO	057	9	100	d
015	SI	058	:	101	e
016	DLE	059	;	102	f
017	DC1	060	<	103	g
018	DC2	061	=	104	h
019	DC3	062	>	105	i
020	DC4	063	?	106	j
021	NAK	064	@	107	k
022	SYN	065	A	108	l
023	ETB	066	B	109	m
024	CAN	067	C	110	n
025	EM	068	D	111	o
026	SUB	069	E	112	p
027	ESCAPE	070	F	113	q
028	FS	071	G	114	r
029	GS	072	H	115	s
030	RS	073	I	116	t
031	US	074	J	117	u
032	SPACE	075	K	118	v
033	!	076	L	119	w
034	"	077	M	120	x
035	#	078	N	121	y
036	$	079	O	122	z
037	%	080	P	123	(
038	&	081	Q	124	|
039	'	082	R	125	|
040	(	083	S	126	~
041	)	084	T	127	DEL
042	•	085	U		

ASCII codes are in decimal.
LF=Line Feed, FF=Form Feed, CR=Carriage Return, DEL=Rubout.

ตารางที่ 3 ตารางค่า ASCII ของ CP/M

Flash ใน Apple เมื่อบวกค่า ASCII ด้วย 128 ก็จะเป็นชนิด Normal

HGR

หน่วยความจำที่ใช้ในการแสดงผลแบบ High Resolution Graphic นั้นอยู่ที่ address 2000 ถึง 3FFF ใน Standard apple หรือใน apple CP/M ก็เป็น address 1000 ถึง 2FFF การจัดหน่วยความจำก็คล้าย ๆ กับใน Text Screen

ลองทดลอง RUN โปรแกรมที่ 4 (ซึ่งต้อง RUN ด้วย GBASIC) จะช่วยให้เข้าใจยิ่งขึ้น

```
10 HGR 2
20 FOR I = &H1000 TO &H2FFF
30 POKE I, &HFF
40 NEXT I
```

โปรแกรมที่ 4 พิมพ์บน HGR screen

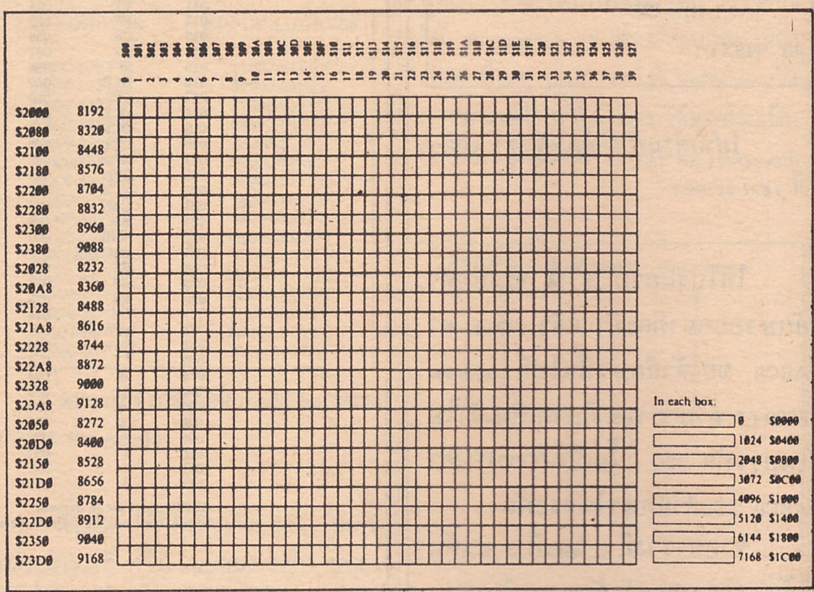
เมื่อ RUN โปรแกรมที่ 4 แล้ว จะเห็นได้ว่า การจัดหน่วยความจำของ HGR screen คล้าย ๆ กับการจัดหน่วยความจำของ Text Screen แต่จะต่างกันตรงที่ใน HGR Screen หน่วยความจำ 1 Byte สามารถแทนจุดบนจอได้เพียง 7 จุดเท่านั้น และถ้าเป็นจอสีคุณก็จะเห็นได้เพียง 3 หรือ 4 จุดเท่านั้น ซึ่งใน Text Screen หน่วยความจำ 1 Byte สามารถแทนตัวอักษรได้ 1 ตัว

สำหรับจอภาพขาวดำ หรือ จอภาพสีเขียว แต่ละบิทของหน่วยความจำนั้น จะแทนการติดหรือดับของจุด 1 จุด คือถ้าบิทนั้นเป็น 0 จุดก็ดับ ถ้าบิทนั้น เป็น 1 จุดก็ติด

> **"การเล่น Graphic ใน CP/M นี้น่าสนใจและน่าสนุกกว่าการเล่นใน Applesoft BASIC เพราะเราสามารถเล่น Graphic ได้ด้วยภาษา BASIC, ด้วยภาษา Assembly หรือด้วยภาษาอื่น ๆ"**

หน่วยความจำ 1 Byte จะประกอบด้วย 8 บิท แต่ใช้อ้างถึงจุดเพียง 7 จุด ดังนั้น จึงเหลืออีก 1 บิท ที่ไม่ได้ใช้แทนจุด บิทที่เหลือนี้จะทำให้จุดแต่ละจุดนั้นเยื้องกันเล็กน้อย คือเมื่อบิทที่เหลือนี้เป็น 1 จะทำให้จุดที่เหลือ 7 จุดนั้นพิมพ์ที่ตำแหน่งหนึ่ง แต่เมื่อบิทที่เหลือเป็น 0 จะทำให้จุดทั้งเจ็ดนี้แต่ละจุดเยื้องไปจากตำแหน่งเดิมเล็กน้อย ถ้านับเป็นหน่วยก็คงวัดได้ประมาณครึ่งจุด

ทีนี้ บิทแต่ละบิทนั้นแทนตำแหน่งไหนของจุด ก็คงต้องดูรูปที่ 4 ซึ่งจะเห็น



รูปที่ 3 address ของ High Resolution Graphic Screen

ได้ว่า **บิทที่มีน้ำหนักน้อยที่สุด จะแทนจุดข้างซ้ายสุด และบิทที่ 6 ก็จะแทนจุดทางขวาสุด** ส่วนอีกบิท ซึ่งมีค่ามากที่สุด ก็เป็นบิทที่เหลือไป

LSB	0	1	2	3	4	5	6	7	MSB
	0	0	●	0	●	●	0		
	0	0	1	0	1	1	0	0	= $(34)_{16}$

รูปที่ 4 แสดงการแทนจุดแต่ละจุดในหน่วยความจำ

จากรูปที่ 4 จะเห็นได้ว่า มีจุดที่ 2. 4 และ 5 ที่ติด ดังนั้นบิทที่ 2, 4 และ 5 จึงเป็น 1 ส่วนบิทที่ 7 จะเป็น 0 หรือ 1 ก็ได้ สมมติว่าให้เป็น 0 ดังนั้นหน่วยความจำซึ่งแทน 7 จุดนี้ จึงมีค่าเท่ากับ 34 ฐาน 16 (แต่ละบิทถูกเรียงจากค่าน้อยไปมาก ดังนั้น การเปลี่ยนเป็นฐาน 16 ต้องอ่านจากขวาไปซ้าย)

ถ้าคุณเล่น Graphic ด้วย GBASIC คุณก็สามารถใช้คำสั่ง HGR, HPLOT หรือคำสั่งอื่น ๆ ได้เช่นเดียวกับใน Applesoft BASIC แต่คุณก็สามารถเล่น Graphic ได้มากกว่านี้ ด้วยการใช้คำสั่ง POKE ซึ่งเป็นคำสั่งที่เก็บค่าลงหน่วยความจำโดยตรง ยกตัวอย่างเช่น คุณต้องการพิมพ์ตัวอักษรภาษาไทย บน HGR Screen เป็นต้น

การพิมพ์ตัวอักษรซึ่งเราออกแบบขึ้นมาเองบน HGR Screen ถ้าเราใช้คำสั่ง HPLOT ผลที่ได้จะช้ามาก แต่ถ้าเราใช้คำสั่ง POKE ก็จะช่วยให้เร็วขึ้นมาก เพราะเราเพียง POKE 8 ครั้ง เราก็จะได้ตัวอักษร 1 ตัว เราลองมาพิมพ์ตัวอักษรดูสักตัว สมมุติว่าเป็น ก.ไก่

จากรูปที่ 5 เราจะเปลี่ยนรูป ก.ไก่มาเป็น เลขฐานสอง ฐาน 16 แล้วก็ฐาน 10 ตามลำดับ (เปลี่ยนจากฐานสองเป็นฐาน 16 ต้องกลับจากขว้ามาซ้าย และบิท 7 เป็น 0) เราก็จะได้ข้อมูลที่จะไปพิมพ์แล้ว

ทดลอง RUN โปรแกรมที่ 5 แล้วคุณจะเห็น ก.ไก่บนจอ แต่อย่าลืมสวิทช์

0●●●●●0	0111110	3E	62
●0000000●	1000001	41	65
0●000000●	0100001	42	66
●0000000●	1000001	41	65
●0000000●	1000001	41	65
●0000000●	1000001	41	65
●0000000●	1000001	41	65
●0000000●	1000001	41	65
●0000000●	1000001	41	65
	ฐาน 2	ฐาน 16	ฐาน 10

รูปที่ 5 ตัวอักษร ก.ไก่

```
5   HGR
10  FOR I = 0 TO 7
20  READ DOT (I)
30  NEXT I
40  DATA 62, 65, 66, 65, 65, 65,
    65, 65
50  FOR I = 0 TO 7
60  POKE 4096 + (1024*I), DOT (I)
70  NEXT I
```

โปรแกรม 5 พิมพ์ ก.ไก่ บน HGR Screen

ไป 40-Column เพราะคุณจะไม่เห็นอะไรเกิดขึ้นบน 80-Column เลย

ทีนี้ ถ้าคุณอยากทดลองความเร็วของมัน ก็ลองพิมพ์ดูให้เต็มบรรทัดหรือเต็มจอก็ได้ เทียบกับวิธีใช้คำสั่ง HPLOT ดู

ตอนนี้ คุณคงพอมองเห็นแนวความคิดที่จะพิมพ์ตัวอักษรภาษาไทยบน Graphic Screen แล้ว ซึ่งการทำภาษาไทยแบบที่สามารถพิมพ์ได้ 24 บรรทัดในจอภาพเดียว เขาก็ใช้ Graphic Screen กัน แต่สำหรับ Apple นั้น Graphic ยังค่อนข้างหยาบ การทำ 24 บรรทัดในจอภาพเดียวคงทำได้ยาก แต่ถ้าคุณยอมลดจำนวนบรรทัดลงก็พอจะทำได้ และถ้าคุณใช้ภาษา Assembly หรือภาษาเครื่องในการพิมพ์แล้ว ความเร็วในการพิมพ์ก็จะสูงขึ้นมากจนไม่ไม่ต่างจากการพิมพ์ตัวอักษรใน Text Screen ปกตินัก

ASSEMBLY

การเล่น Graphic ด้วยภาษา assembly จะได้ภาพที่เคลื่อนไหวเร็วมาก

SCREEN SOFT SWITCHES

Location Hex	Decimal		Description:
$E050	57424	- 8112	Display a GRAPHIC mode.
$E051	57425	- 8111	Display TEXT mode.
$E052	57426	- 8110	Display all TEXT or GRAPHICS.
$E053	57427	- 8109	Mix TEXT and a GRAPHICS mode .
$E054	57428	- 8108	Display the Primary page (Page 1)
$E055	57429	- 8107	Display the Secondary page (Page 2)
$E056	57430	- 8106	Display LO-RES GRAPHICS mode.
$E057	57431	- 8105	Display HI-RES GRAPHICS Mode.

Screen Mode Combinations

Primary Page			Secondary Page		
Screen	Switches		Screen	Switches	
All Text	$E054	SE051	All Text	$E055	$E051
All Lo-Res	$E054	SE056	All Lo-Res	$E035	$E056
Graphics	$E052	SE050	Graphics	$E052	$E050
All Hi-Res	$E054	SE057	All Hi-Res	$E055	$E057
Graphics	$E052	SE050	Graphics	$E052	$E050
Mixed Text	$E054	SE056	Mixed Text	$E055	$E056
and Lo-Res	$E053	SE050	and Lo-Res	$E053	$E050
Mixed Text	$E054	SE057	Mixed Text	$E055	$E057
and Hi-Res	$E053	SE050	and Hi-Res	$E053	$E050

ตารางที่ 4 ตารางตำแหน่งหน่วยความจำที่ใช้ในการสวิทช์ Graphic

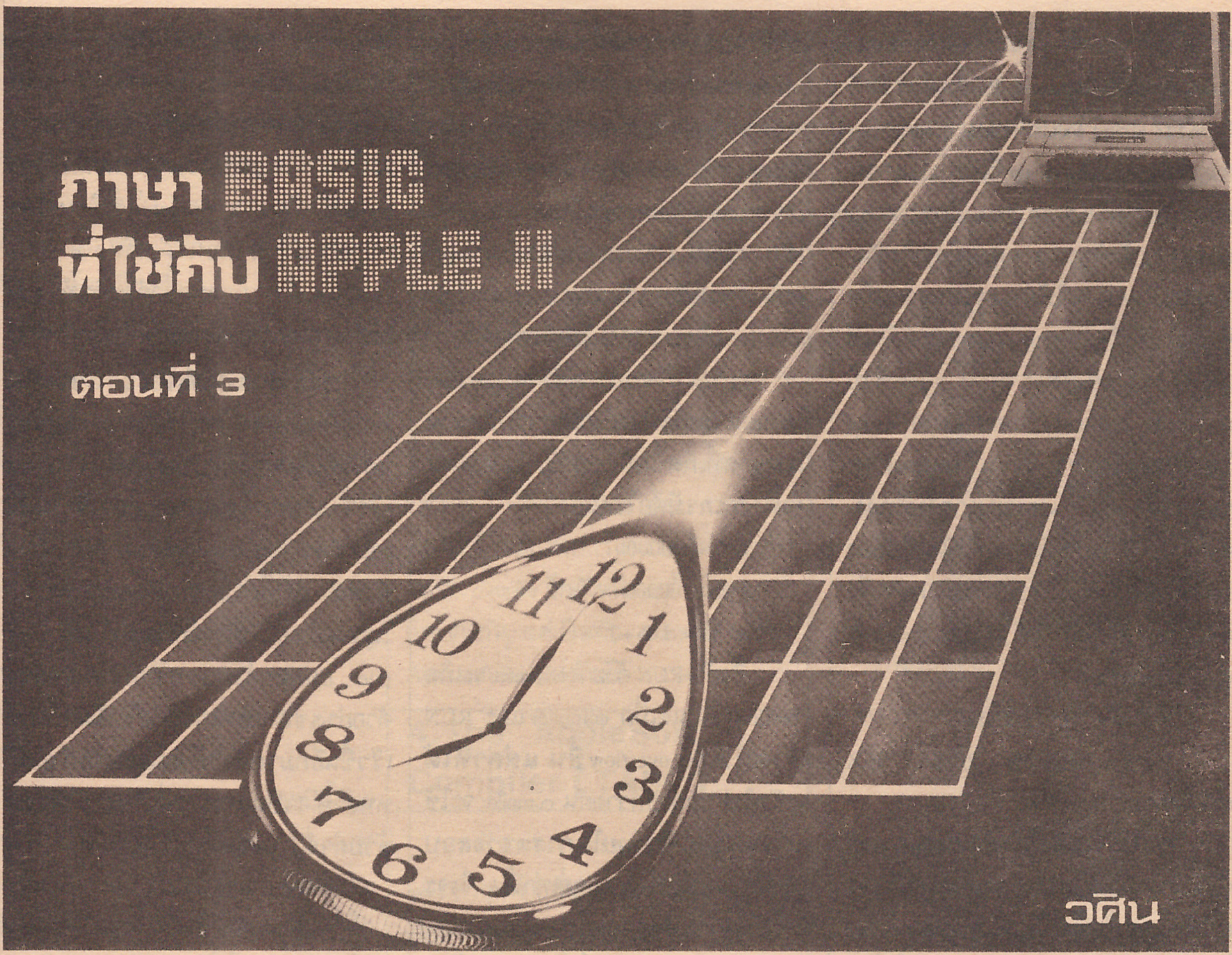
กว่าการเขียนด้วยภาษา BASIC มาก แต่ข้อเสียคือ เราต้องสร้าง Routine ต่าง ๆ ขึ้นใช้เอง การสวิทช์เกี่ยวกับ Graphic ก็ต้องทำเอง ซึ่งการสวิทช์ก็ทำได้โดยอ่านหรือย้ายค่าไปยังหน่วยความจำที่กำหนดในตารางที่ 4

ตัวอย่างเช่น ถ้าเราต้องการเล่น High Resolution Graphic แบบเต็มจอภาพก็ทำได้โดย MOVE ค่าจาก E504 E052, E057, E050 แล้วโยกสวิทช์หรือเปลี่ยนไปที่ 40-Column จอภาพก็จะว่างเปล่าพร้อมให้เราเล่น Graphic ทีนี้เมื่อเราMOVE ค่าไปยังหน่วยความจำของ HGR Screen คือที่ address 1000 ถึง 2000 ก็จะมีจุดปรากฏบนจอทันที เช่นตัวอย่างโปรแกรมที่ 6

```
START  ORG  100H
       STA  OE054H
       STA  OE052H
       STA  OE057H
       STA  OE050H
       STA  A, 3EH
       STA  1000H+0
       MVI  A, 41H
       STA  1000H+400H
       MVI  A, 42H
       STA  1000H+800H
       MVI  A, 41H
       STA  1000H+OCOOH
       STA  1000H+1000H
       STA  1000H+1400H
       STA  1000H+1800H
       STA  1000H+1C00H
LOOP   JMP  LOOP
       END  START
```

โปรแกรมที่ 6 พิมพ์ ก. ไก่ ด้วย assembly

จากตั้งแต่ต้น คุณจะเห็นได้ว่าใน CP/M เราก็สามารถทำ Graphic ได้เช่นกัน อีกทั้งยังสามารถพลิกแพลงได้มากกว่าใน Standard Apple อีกด้วย แต่อุปสรรคที่โดยมากพบกันคือ ความไม่เข้าใจในข้อแตกต่างระหว่างการใช้ CP/M กับ Standard Apple ซึ่งผู้เขียนหวังว่าบทความนี้คงมีประโยชน์สำหรับผู้ที่สนใจศึกษาวิชาการด้านนี้อย่างจริงจังบ้าง ไม่มากก็น้อย ■



(ต่อจากฉบับที่แล้ว)

15. TRACE/NOTRACE ต้องใช้ /D ในการ compiler จึงจะมีการทำงานตามคำสั่งทั้งสองไม่เช่นนั้น compiler จะข้ามคำสั่งทั้งสองนี้ไปและแจ้งให้ทราบเป็นการเตือน (warning message) ในตอน compile

16 USR Function

สามารถใช้ USR Function กับ compiler ได้ก็จริงแต่ไม่สามารถส่ง parameter ไปให้โปรแกรมย่อยที่เป็นภาษาเครื่องซึ่ง Define ไว้เป็น USR Function ได้อย่างใน interpreter ดังนั้นการส่งผ่าน parameter อาจทำได้โดยวิธีอื่นคือ

1. PEEK ค่าของ parameter ที่ต้องการจากโปรแกรมหลักและ POKE เข้าไปเก็บไว้ใน memory ส่วนที่จัดเอาไว้พิเศษเพื่อไม่ให้ไปทับโปรแกรมหรือ data ส่วนอื่น การจัดที่ไว้เป็นพิเศษนี้อาจทำได้โดยการแก้ไขข้อมูลใน file BCLOAD ซึ่งบอก Address ที่จะ load โปรแกรมเข้าไปต่อจาก Runtime Module โดยเลื่อนแอดเดรสให้สูงขึ้น ทำให้มีช่วงว่างระหว่าง-Runtime Module กับตัวโปรแกรมหลัก ซึ่งเนื้อที่ของ memory ในช่วงนี้แหละที่จะใช้ POKE ค่าของ parameter และอาจรวมถึงตัวโปรแกรมภาษาเครื่องเข้าไปด้วยเลยก็ได้

2. ถ้าโปรแกรมภาษาเครื่องสั้นพอคุณอาจแปลงจากภาษาเครื่องในรูปเลขฐานสิบหกให้กลายเป็น ASCII Code และกลายเป็นตัวอักษรต่าง ๆ ใน string แต่จะต้องเป็น string **แรกที่ถูกกำหนดในโปรแกรมนั้นด้วย** โดยแทรก (insert) string เข้าไปทีละตัวเช่นคำสั่ง MOV B,C ของ 8080 มี code 41 ฐานสิบหก (65 ฐานสิบ เท่ากับตัว A ของ ASCII Code) ก็แทรกค่า ASCII code 65 เข้าไปใน string ณ ตำแหน่งที่ต้องการ โดยใช้ฟังชัน-CHR$ (65) ดังนั้นโปรแกรมและ data จะถูกเก็บในเนื้อที่สำหรับ string (string space)

การหาแอดเดรสตั้งต้นของโปรแกรม (ในรูป string) เพื่อเรียกใช้นั้น

ทำได้โดย PEEK ค่าจาก pointer ของ-string ดังนี้คือ ใน compiler นั้นการจัดที่สำหรับ string ไม่เหมือนใน interpreter สมมุติว่า A$เป็น string ที่เก็บโปรแกรมย่อยภาษาเครื่องและ data เมื่อจะค้นหาแอดเดรสตั้งต้นของ string ด้วยฟังชัน-VARPTR (A$) จะให้ค่าแอดเดรสค่าหนึ่งซึ่งไม่ใช่ที่เก็บ string โดยตรงแต่เป็นที่เก็บข้อมูลกำหนดการจัดรูป string ซึ่งมีสามไบท์ ไบท์แรกเป็นความยาว string ไบท์ที่สองและสามจึงจะเป็นแอดเดรสของ string จริง ๆ โดยเรียงจากไบท์หลัง-(least significant) ขึ้นก่อน ดังนั้นเมื่อต้องการหาจุดเริ่มต้นโปรแกรมดังกล่าวจึงต้องใช้คำสั่งดังนี้ (ให้ A$เป็น string ที่เก็บ โปรแกรม)

```
I % =VARPTR (A$)
AD% = PEEK (I% + 2) *256+PEEK
(I%+1)
```

จะได้ AD% มีค่าเป็นแอดเดรสที่ต้องการ แต่เนื่องจาก string อาจมีการย้ายตำแหน่งไปภายใน string space ดังนั้นโปรแกรมภาษาเครื่องของ Z-80 จึงต้องใช้ JUMP แบบ relative

การส่ง parameter กลับของฟังชัน USR ทำได้โดยการให้โปรแกรมภาษาเครื่องใส่ค่าลงใน Register pair HL

7. WHILE/WEND ต้องเป็น statically nested เช่นเดียวกับ FOR/NEXT คือมี WEND เพียงตัวเดียวเท่านั้นที่คู่กับ WHILE 1 ตัว และ loop จะต้องไม่คาบเกี่ยวกันหรือคาบเกี่ยวกับ loop ของ FOR/NEXT

3. ความแตกต่างอื่น มีดังนี้คือ

3.2 การคำนวณค่าจากนิพจน์ทางคณิตศาสตร์ (expression evaluation) ในกรณีที่นิพจน์ประกอบด้วยตัวแปรหลายชนิด BASIC Compiler จะทำการคำนวณโดยตัดสินเรื่องชนิดของตัวแปรที่จะใช้ในการคำนวณตั้งแต่ตอน Compile และจะแปลงตัวแปรที่มีความละเอียดของตัวเลขต่ำกว่าให้เป็นชนิดที่มีความละเอียดสูงกว่า เช่น Single precesion แปลงเป็น double precision เช่น ANS=J% +A!+Q#ในกรณีนี้ J % จะถูกแปลงเป็น single prlcision บวกกับ A ! และแปลงเป็น double precision เพื่อบวกกับ Q#อีกทีหนึ่ง

อย่างไรก็ตามการใช้งาน compiler มีความสามารถในการจัดการกับความผิดพลาดที่เกิดขึ้นน้อยกว่า interpreter เช่นโปรแกรมต่อไปนี้

```
I % = 20000
J % = 20000
M % = I %+J %
```

ซึ่งถ้า RUN ด้วย interpreter จะเกิด overflow ขึ้น แต่เมื่อ compiler และ RUN แล้วจะไม่เกิด overflow **ขึ้น แต่ค่าที่ได้จะผิดไป** นอกจากว่าตอน compile จะใช้ /D (Debug) เพื่อช่วยในการตรวจสอบการทำงานของโปรแกรมเท่านั้น จึงจะทำให้สามารถตรวจพบการ overflow และรายงานผลออกมาให้ทราบ พร้อมทั้งหยุดการทำงานของโปรแกรม

ข้อควรระวังอีกประการหนึ่งก็คือ compiler จะทำงานโดยการ optimize คือพยายามจัดโปรแกรมที่แปลให้มีประสิทธิภาพสูงสุด ดังนั้นในบางกรณีอาจเกิดความผิดพลาดที่คาดไม่ถึงได้เช่น โปรแกรมต่อไปนี้อาจให้ผลลัพธ์ที่ไม่ถูกต้อง ได้ คือ

```
E%=20000
J%=-18000
K%=20000
M%=I% + 9% + K%
```

ซึ่งในการ compile ถ้า compiler จัดลำดับการคำนวณให้ I% บวกกับ J% ก่อนก็จะไม่เกิด overflow แต่ถ้า I% บวกกับ K% ก่อนก็จะ overflow ดังนั้นจึงควรระลึกว่า compiler จะทำงานตามลำดับความสำคัญของ operator หรือเครื่องหมายเช่น * มาก่อน + หรือ/ มาก่อน - เท่านั้น ถ้าจะกำหนดลำดับการทำงานให้ชัดเจนลงไปจะต้องใส่วงเล็บกำกับด้วย

3.2 ตัวแปรที่เป็นจำนวนเต็ม (integer variable) การใช้ตัวแปรที่เป็นจำนวนเต็มจะทำให้โปรแกรมทำงานรวดเร็วขึ้นมาก โดยเฉพาะอย่างยิ่งในตัวแปรที่ควบคุม loop และในการคำนวณตัวพ่วง (subscript) ของตัวแปรแต่ละตัวใน Array และยังทำให้ภาษาเครื่องที่ Compile ออกมาสั้นและกระทัดรัดอีกด้วย ตัวอย่างเช่น โปรแกรมต่อไปนี้จะทำงานเร็วขึ้นอีกมากถ้าใช้ I% แทน I หรือ กำหนดให้ I เป็นตัวแปรที่เป็นจำนวนเต็มด้วยคำสั่ง DEFINT

```
FOR I = 1 TO 10
A(I) = 0
NEXT I
```

3.3 ฟังชั่นทางคณิตศาสตร์ที่เป็น Double Precision ในการใช้ compiler คุณสามารถใช้ตัวแปรหรือค่าคงที่ที่ double precision เป็น operand ของฟังชั่นต่าง ๆ ได้ รวมทั้งฟังชั่น Trancendental (SIN,COS,TAN,ATN,LOG,EXP,SQR) ด้วย ซึ่งต่างจากใน interpreter ที่ใช้ได้เฉพาะ single precision เท่านั้น ดังนั้นในการเขียนและตรวจสอบโปรแกรมจึงต้องใช้ทุกค่าที่คุณต้องการให้เป็น double precision เป็นรูป single precision ก่อนเพื่อให้ RUN กับ interpreter ได้ เมื่อจะ compile จึงค่อยแก้ค่าเหล่านั้นให้เป็นไปตามที่ต้องการอีกทีหนึ่ง ซึ่งเมื่อ compile แล้วความสามารถดังกล่าวก็จะ

ถูกรวมเข้าในโปรแกรมด้วย

3.4 ตัวแปรควบคุม loop ที่เป็น Double Precision สามารถใช้ตัวแปรควบคุม loop ที่เป็น Double Precision ได้ทำให้สามารถเพิ่ม Step ในคำสั่ง FOR/NEXT ได้ละเอียดขึ้น

3.5 การจัดที่ว่างสำหรับเก็บ string (string space) ของ compiler จัดต่างกับ interpreter การใช้คำสั่ง POKE ฟังชัน PEEK หรือ VARPTR ตลอดจนโปรแกรมย่อยภาษา Assembly ที่มายุ่งกับบริเวณที่เก็บข้อมูลในการจัดรูป string (string description) อาจทำให้เกิด String Space Corrupt Error ขึ้นได้ (ลองดูในตอนที่อธิบายถึง ฟังชัน USR)

รายงานความผิดพลาด (Error Messages)

รายงานความผิดพลาดจะมีทั้งในช่วงการ compile และในช่วงการ RUN ในช่วง compile ความผิดพลาดที่เกิดขึ้นก็แบ่งได้เป็น 2 ชนิดคือ Fatal Error คือความผิดพลาดร้ายแรงซึ่งโปรแกรมจะไม่สามารถทำงานได้ กับ warning Error คือความผิดปกติที่ compiler ตรวจพบ และอาจเป็นสิ่งผิดพลาดได้ compiler จึงแจ้งให้ทราบเป็นการเตือน การรายงานความผิดพลาดในช่วงการ compile นี้-compiler จะแสดง line ที่เกิด error ลูกศรเหนือตำแหน่งที่เกิด error ใน line นั้น และ error code ที่เป็นอักษร 2 ตัว

ในบางครั้ง compiler จะอ่าน line นั้นเลยตำแหน่งที่เกิด error ไปบ้างเพื่อให้แน่ใจว่าเกิด error ขึ้นจริง ดังนั้นลูกศรอาจชี้เลยตำแหน่งที่เกิดความผิดพลาดไปบ้างหรือชี้ที่สุด line นั้น

ส่วนรายงานความผิดพลาดที่เกิดขึ้นในช่วงการ RUN จะมีการแจ้งลักษณะความผิดพลาดเป็นข้อความตามด้วยแอดเดรสที่เกิดขึ้น และถ้ามีการใช้ /D,/E หรือ /X ในตอน compiler จะมีการแจ้ง line number ใน source โปรแกรมที่เกิด error ขึ้นด้วย รหัสของความผิดพลาด (error code) ในช่วงการ run นี้จะตรงกับรหัสที่ใช้กับ interpreter แต่มีไม่ครบ โดยเว้นข้ามบางตัวไป เนื่องจากความผิดพลาดเหล่านั้นจะถูกตรวจพบตั้งแต่ตอน compile แล้ว ไม่มีทางหลุดรอดมาเกิดขึ้นในตอน RUN

ครับ นั่นก็เป็นเรื่องราวของ BASCOM ซึ่งเป็น BASIC compiler ของบริษัท ไมโครซอฟท์ที่ใช้กับ CP/M ซึ่งต่อไปนี้ จะเป็นการแนะนำให้รู้จักกับ BASIC compiler อีกตัวหนึ่งของไมโครซอฟท์อีกเช่นกัน แต่เป็นตัว compiler ภาษา Applesoft BASIC นั่นคือ TASC

4. The AppleSoft Compiler (TASC)

ตัวแปลภาษา BASIC แบบ Compiler ตัวสุดท้ายที่จะพูดถึงก็คือ TASC ซึ่งย่อมาจาก The AppleSoft Compiler ซึ่งพัฒนาโดยบริษัทไมโครซอฟท์ อีกเช่นเดียวกัน แต่จะแปลภาษา AppleSoft BASIC ให้เป็น machine language เลย

ที่จริงแล้ว interpreter ของ AppleSoft ใน ROM ที่มากับเครื่อง Apple ทุกเครื่องนั้นที่จริงแล้วก็พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ เช่นเดียวกัน บริษัท Apple Computer ได้นำไปดัดแปลงในเวลาต่อมาและนำไปใช้กับเครื่อง Apple II และกลายเป็นภาษา BASIC ยอดนิยมอีกภาษาหนึ่งไป

TASC จะแปลภาษา AppleSoft BASIC ให้เป็น machine code ซึ่งจะทำให้ความเร็วในการทำงานเร็วขึ้น (ระหว่าง 2 ถึง 20 เท่า แล้วแต่โปรแกรมที่นำมา Compile) กว่าเมื่อ RUN ภายใต้ interpreter นอกจากนี้ยังมีข้อดีคือ สามารถทำการคำนวณแบบ interger ได้เช่นเดียวกับ Integer BASIC ทำให้ทำงานเร็วขึ้น

Package ของ TASC ประกอบด้วย file ต่างๆ ดังนี้คือ

1. TASC คือตัวโปรแกรม Compiler เอง
2. PASSO, PASS1, PASS2 เป็นส่วนประกอบภายในของ TASC
3. RUNTIME เป็นห้องสมุดโปรแกรมหรือ Library เป็นที่เก็บ Subroutine ที่เป็น machine code สำหรับคำสั่งต่างๆ

   ส่วนของ RUTIME นี้จะถูก object program เรียกใช้ในระหว่างการ execution
4. ADR เป็น Utility program สำหรับใช้กับ binary file
5. CREATEADR เป็นโปรแกรมสำหรับ create binary file
6. BALL เป็นตัวอย่างโปรแกรมภาษา Applesoft BASIC

ตัวอย่างการใช้งาน

ในการใช้งาน TASC จะเริ่มจากการ RUN file TASC ซึ่งเป็นโปรแกรมที่อยู่ในรูป Applesoft BASIC ดังนี้ (ในตัวอย่างนี้เราจะใช้โปรแกรมตัวอย่าง BALL)

RUN TASC<RETURN>

SOURCE FILE? BALL <RETURN>

(ถามชื่อ SOURCE file ที่ต้องการ)

OBJECT CODE FILE:

(DEFAULE BALL OBJ)? <RETURN>

ถามชื่อ object file ที่ต้องการ)

โปรแกรมที่ระบุชื่อจะต้องอยู่บนดิสค์อยู่แล้ว ถ้าจะระบุว่าอยู่ที่ drive หรือ slot อื่น ก็ใช้ Sn, Dn ต่อท้ายชื่อ file ไปได้เลย (n คือเลขที่ของ drive หรือ Slot ที่ต้องการ) ส่วนชื่อ object file (binary file) ที่จะได้จากการ compile นี้เครื่องจะตั้งให้ (default) เป็นชื่อเดิม โดยมีคำว่า OBJ ต่อท้าย ถ้าไม่ระบุเป็นอย่างอื่น

ในระหว่างการทำงาน จะมีการ access disk ค่อนข้างสม่ำเสมอ ทั้งเมื่ออ่าน source program และสร้าง object code เป็น file ใหม่บนดิสค์ ระหว่างนั้นเครื่องจะถามว่า

MEMORY USAGE : ถามว่าจะกำหนดการใช้งาน memory หรือไม่ (จะอธิบายทีหลัง)

DEFAULT CONFIGURATION?

<RETURN>

OPTIONS : จะเลือก option ใดหรือไม่ (จะอธิบายทีหลัง)

DEFAULT CONTIGURATION?

<RETURN>

ทั้ง 2 คำถามนี้ถ้าไม่ตอบ เครื่องจะจัดให้ตามที่ default ไว้ การทำงานของ TASC นี้จะแบ่งเป็น 2 ขั้นตอนคือ ขั้นการวิเคราะห์ภาษาและสร้าง object code (Syntax Analysis & Code Generation) เป็นขั้นแรกหรือ PASS1 ส่วนขั้นที่สองหรือ PASS2 คือการจัดที่ในหน่วยความจำให้ตัวแปรต่างๆ (Variable Allocation)

ในขณะที่ทำการ Compile ใน PASS1 source program ส่วนที่กำลังถูก compile จะถูกพิมพ์ออกมาทางจอไปด้วย และหากมี error เกิดขึ้นก็จะพิมพ์บอกออกมาทันที ใน PASS1 นี้จะมีการ generate object code เกือบทั้งหมดไปเรื่อยๆ จะขาดก็แต่แอดเดรสของตัวแปรต่างๆ ซึ่งยังไม่สามารถกำหนดได้จนกว่าจะ Compile จบทั้งโปรแกรม ดังนั้นใน object code ที่ได้ตอนนี้จึงเว้นที่ไว้เพื่อให้เติมแอดเดรสลงไปทีหลังใน PASS2

เมื่อ Source program ทั้งหมดถูก compile แล้วก็เป็นอันจบ PASS1 เครื่องจะพิมพ์ออกมาว่า

*****BEGINING PASS2

ซึ่งจะเป็นการเติมแอดเดรสต่างๆ ที่เว้นไว้ลงใน object code ที่ได้หรือเรียกว่าการจัดที่ให้แก่ variable ต่างๆ นั่นเอง เมื่อกระบวนการนี้สิ้นสุดลง เครื่องจะพิมพ์ข้อความว่า

*****CODE GENERATION COMPLETE

COMPILATION INFORMATION AND LINE NUMBER REFERENCE TABLE? YES

ถ้าตอบ YES คือต้องการ listing จากการ compile ซึ่งจะออกทางจอ หรือถ้าจะให้ออกทาง printer ก็ใช้คำสั่งคือ CTRL-D PR # Printerslot และเมื่อพิมพ์ listing เสร็จแล้วก็จะกลับสู่ Applesoft interpreter ด้วยการแสดงเครื่องหมาย]ตามเดิม

*****COMPILATION COMPLETE

ทีนี้ถ้า CATALOG ในแผ่นดิสค์ออกมาดูก็เห็นว่ามี file BALL.OBJ เพิ่มขึ้นมา นั่นคือ object file นั่นเอง ซึ่งการ RUN ก็ทำได้ดังนี้

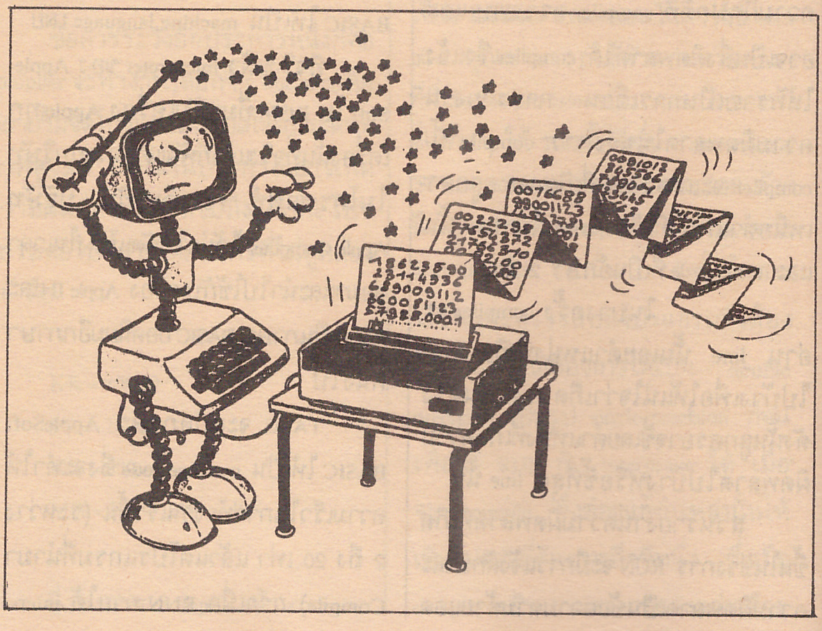
)BLOAD RUNTIME

)BRUN BALL.OBJ

โปรแกรมก็จะทำงาน ซึ่งเมื่อเปรียบเทียบกับการ RUN Source program BALL ด้วย interpreter แล้วจะเห็นว่าเร็วกว่ามาก

การทำงานของ TASC

เมื่อเปรียบเทียบกับ CP/M BASIC Compiler (MBASIC และ GBASIC) แล้วจะเห็นว่าการใช้งาน TASC ง่ายกว่าและไม่ต้องมีการ link เลย ที่เป็นเช่นนี้เพราะโครงสร้างและการทำงานภายใน TASC



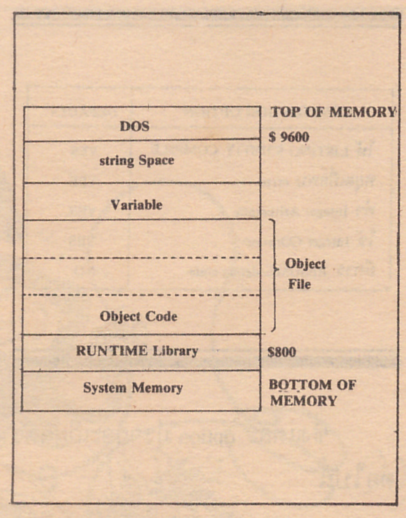
เป็นดังนี้

เมื่อสั่ง RUN TASC โปรแกรม TASC ซึ่งเป็น Applesoft BASIC ที่ RUN ได้ interpreter จะทำงานโดยเรียก object program PASS0 มาทำงาน โดยเตรียม parameter ต่างๆ ของโปรแกรมสำหรับการทำงานใน PASS1 และ PASS2 ต่อไป จากนั้นการ compile จริงๆ จึงจะเริ่มต้นโดย PASS1 จะทำหน้าที่วิเคราะห์ภาษา (Syntax analysis) และสร้าง object code เท่าที่จะทำได้ แต่ยังไม่กำหนดแอดเดรสลงไปแน่นอน (เทียบได้กับสภาพ Relocatable file หรือ .REL ของ CP/M นั่นเอง) ส่วน PASS2 จะทำการจัดที่โดยกำหนดแอดเดรสให้กับตัวแปร (variable) ต่างๆ การทำงานทั้ง 2 PASS นี้จะใช้เวลาค่อนข้างสั้นเพราะไม่ต้องมีการ link เนื่องจากมี RUNTIME Library เข้าช่วยแบบเดียวกับ MBASIC และ GBASIC ที่มี Runtime module นั่นเองคือเป็น Routine ย่อยที่กำหนดแอดเดรสไว้แน่นอนแล้ว นอกจากนี้ยังอาจเรียกใช้บาง subroutine ได้จาก interpreter ใน ROM อีกด้วย ดังนั้นจึงเพียงแต่จัดการเรื่องแอดเดรสของตัวแปรเท่านั้น ส่วนแอดเดรสของ Subroutine หรือโปรแกรมย่อยสำหรับคำสั่งต่างๆ นั้นจะสามารถกำหนดให้ได้ตั้งแต่ตอน Code Generation ใน PASS1 เลยเพราะแน่นอนอยู่แล้ว (ซึ่งต่างกับ Runtime Module ของ CP/M BASIC Compiler ที่จะมี Subroutine ที่ใช้กันมาก**เกือบทั้งหมด**เท่านั้น ไม่ใช่ทั้งหมดอย่าง TASC และเวลา link นั้นก็จะไปหา Subroutine นอกเหนือจากใน Runtime Module จาก BASLIB. อีกทีหนึ่งถ้าจำเป็น (ดูในฉบับที่ 8)

การใช้ RUNTIME library แทนการ link ทั้งหมดอย่างที่ใช้ใน TASC นี้มีข้อดีคือทำให้การ compile ง่ายและสะดวกไม่ยุ่งยาก และส่วน object code ที่ได้มีขนาดเล็ก (เพราะไปเรียกใช้ subroutine ใน RUNTIME และบางส่วนของ interpreter ใน ROM แทน) ทำให้ประหยัดเนื้อที่ในดิสค์ เพราะแทนที่ทุกโปรแกรมจะต้องรวมเอา subroutine ที่จำเป็นเข้าไว้ด้วยในโปรแกรม ซึ่งจะซ้ำกันและเปลืองที่โดยไม่จำเป็น ก็แยกเอาส่วน subroutine นั้นออกมาเป็น file ต่างหากที่ทุกโปรแกรมเรียกใช้ได้ร่วมกันใน RUNTIME Library แทน อย่างไรก็ตาม มีข้อที่อาจพิจารณาเป็นข้อเสียได้ประการหนึ่ง คือ การที่ RUNTIME Library ทั้งหมดจะต้องถูกใช้พร้อมๆ กัน คือต้อง load ทั้งหมดเข้าไปใน memory ถึงแม้โปรแกรมจะใช้เพียงบาง subroutine เท่านั้นก็ตาม เช่นโปรแกรมที่ไม่ใช้ string เลย ก็ยังต้องมี subroutine ที่จัดการเรื่อง string อยู่ใน memory ด้วยเป็นต้น แต่โดยทั่วไปแล้วโปรแกรมขนาดใหญ่จะใช้ subroutine ใน RUNTIME Library ทั้งหมด ส่วนโปรแกรมเล็กๆ ก็จะมีที่เหลือพอสำหรับ RUNTIME Library โดยไม่ก่อให้เกิดปัญหาแต่อย่างใด

ในระหว่างการ RUN โปรแกรมที่ได้จากการ Compile โดย TASC ถ้า RUNTIME Library ยังไม่อยู่ใน Memory ก็ต้อง BLOAD (เพราะเป็น binary file) เข้ามาก่อน แต่ถ้ามีการ RUN โปรแกรมอื่นไปก่อนและเคย BLOAD RUNTIME เข้ามาแล้วก็ไม่ต้องทำอีก เพียงแต่ BRUN object file ที่ต้องการได้เลย

ในระหว่าง execute (หรือ Runtime) ตัว RUNTIME MODULE จะอยู่ใน memory ส่วนใต้ส่วนสำหรับ Hi-res screen ดังนั้นการจัด memory จะเป็นดังรูปที่ 1



รูปที่ 1 "การจัด memory ขณะทำการ execute"

Memory Usage

จากลักษณะการจัด Memory ดังรูปที่ 1 ในบางกรณีจะมีปัญหาเกิดขึ้นเนื่องจาก Screen memory สำหรับ High Resolution graphic นั้นจะเริ่มต้นต่อจาก RUNTIME LIBRARY พอดี ซึ่งจะไปทับกับ object code ของโปรแกรม เพราะตามที่ default ไว้ object code จะเริ่มต่อจาก RUNTIME LIBRARY ทันทีเช่นเดียวกัน ดังนั้น ถ้าโปรแกรมมีการใช้คำสั่ง HGR หรือ HGR2 จะต้องเลื่อน object code ขึ้นไปอยู่ถัดจาก Screen memory ของ HGR หรือHGR2 โดยกำหนดในการตอบคำถามเรื่อง Memory Usage ในตอน Compile ด้วยคำว่า HGR1 หรือ HGR2 ซึ่งจะเลื่อน object code ให้ขึ้นไปเริ่มถัดจากHi-res screen ของ HGR หรือ HGR2 เพื่อไม่ให้ทับกับข้อมูลของกราฟฟิค ดังนั้นการจัด memory ในขณะ runtime ก็จะเป็น รูปที่ 2

Options

ในการ compile ยังมี options ให้เลือกอีกหลายอย่าง ซึ่งจะมีการตั้งไว้ (default) ให้เป็นค่าต่างๆ ดังนี้

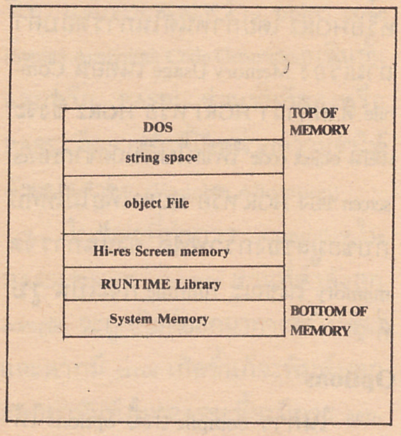
COMPILATION OPTION	DEFAULT
ให้ LISTING จากการ COMPILE	YES
หยุดเมื่อพบ error	YES
ทำ Integer Arithmatic	YES
ใช้ Integer COnstant	YES
มีการ RESUME/debug code	NO

ซึ่งแต่ละ option มีรายละเอียดดังต่อไปนี้

● **listing จากการ compile** ปกติจะถูกสร้างขึ้น ถ้าแก้ไขจาก YES เป็น NO จะไม่มีการสร้าง listing แต่ error, warning หรือ message ต่างๆ จะมีตามปกติ

● **หยุด (pause) เมื่อพบ error** เพื่อเปิดโอกาสให้เลือกว่าจะเลิกการ compile เลย หรือจะให้ compile ต่อ ถ้าเปลี่ยนเป็น NO การ compile จะดำเนินต่อไปจนจบเลย ไม่มีการหยุด

● **Interger Arithmatic** ตามปกติใน Applesoft interpreter (ไม่ใช่ Integer BASIC) ไม่มีการคำนวณแบบ integer การคำนวณส่วนมากจะถูกทำในลักษณะของเลข real ซึ่งกินเนื้อที่ 5 ไบท์ (integer ใช้แค่ 2 ไบท์) ต่อหนึ่งค่า และการคำนวณช้ากว่า integer ประมาณ 2 เท่า (นอกจากค่าบางค่าเท่านั้นที่จะทำแบบ integer เช่น index ที่คอยนับ loop ในคำสั่ง FOR/NEXT) ส่วนตัวแปรที่เก็บค่า integer นั้นจะถูกแปลงเป็น real ในขณะนำไปคำนวณทุกครั้ง ทำให้ช้ากว่าการคำนวณแบบ real ธรรมดาเสียอีก



รูปที่ 2 "**การจัด memory เมื่อมีการใช้** High-Resolution Graphic"

ใน TASC ตัวแปรที่เป็น integer จะถูกนำไปทำการคำนวณแบบ integer เลย ซึ่งเร็วกว่ามากอย่างที่อธิบายแล้ว นอกจากนี้ TASC ยังมีคำสั่ง INTEGER ให้ใช้ในโปรแกรมเพื่อกำหนดว่าตัวแปรไหนจะเป็นแบบ integer อีกด้วย เพื่อที่จะได้ไม่ต้องเติมเครื่องหมาย % ไว้ท้ายตัวแปรทุกตัวที่ต้องการให้เป็น integer

ถ้า Option นี้ถูกเปลี่ยนเป็น NO จะไม่มีการทำ Integer Arithmatic

● **Integer Constant** ค่าคงที่ที่เป็น integer จะถูกเก็บแบบ integer เพื่อให้การคำนวณเร็วขึ้น แต่ถ้ามีการใช้ค่านั้นในการคำนวณทั้งแบบ integer และ real TASC จะสร้าง object code ที่เก็บค่านั้นในทั้ง 2 แบบไว้ เพื่อที่จะได้ไม่ต้องมีการแปลงค่าจาก real เป็น integer หรือ integer เป็น real ในขณะทำการ RUN

การที่เก็บค่าคงที่ในทั้ง 2 แบบจะทำให้ขนาดของ object code ใหญ่ขึ้นเล็กน้อย แต่ตามปกติ option นี้ควรปล่อยให้เป็น YES ไว้นอกจากจะมีปัญหาเรื่องขนาดของ object code ใหญ่เกินไปเท่านั้น

● **RESUME/debug code** option นี้จะใช้เฉพาะในกรณีที่โปรแกรมทำงานแล้วเกิด error ขึ้น โดยจะทำให้ TASC สร้าง opject code พิเศษเพิ่มขึ้นตอนต้นของทุก statement ที่อาจเกิด error ได้ซึ่งจะทำให้ขนาดของโปรแกรมใหญ่ขึ้นและทำงานช้าลงเพราะต้อง execute ส่วนที่เพิ่มเข้ามานี้ด้วย ดังนั้นปกติจึง default เป็น NO ไว้

ถ้าเลือก option นี้เป็น YES จะสามารถใช้คำสั่ง ON ERROR GOTO เพื่อจัดการกับ error และ RESUME เพื่อให้กลับไปทำงานต่อที่ต้น statement เดิมที่ทำให้เกิด error ได้ และเมื่อใช้ option นี้ error message ที่เกิดขึ้นขณะทำการ RUN จะสามารถบอก address ของ object code ที่เกิด error ได้ด้วย

ข้อแตกต่างระหว่าง TASC กับ Applesoft interpreter

ถึงแม้โปรแกรมที่เขียนขึ้นในภาษา Applesoft BASIC ส่วนมากจะสามารถ compile ได้โดย TASC แต่ก็ยังมีข้อแตกต่างหลายประการที่ต้องคำนึงถึง และอาจทำให้เกิดความผิดพลาดขึ้นได้ในการใช้งาน ข้อแตกต่างระหว่างภาษา BASIC ที่ใช้กับตัวแปลภาษาทั้ง 2 มีดังต่อไปนี้

1. คำสั่งเกี่ยวกับการ list หรือ delete source program กับคำสั่งพวก cassette I/O ไม่มีให้ใช้ใน TASC สรุปคำสั่งของ Applesoft BASIC ที่ไม่มีใน TASC ได้ดังต่อไปนี้

CONT DEL LIST LOAD LOMEM: NOTRACE RECALL SAVE SHLOAD STORE TRACE &

2. คำสั่งต่อไปนี้มีการใช้งานที่แตกต่างกัน

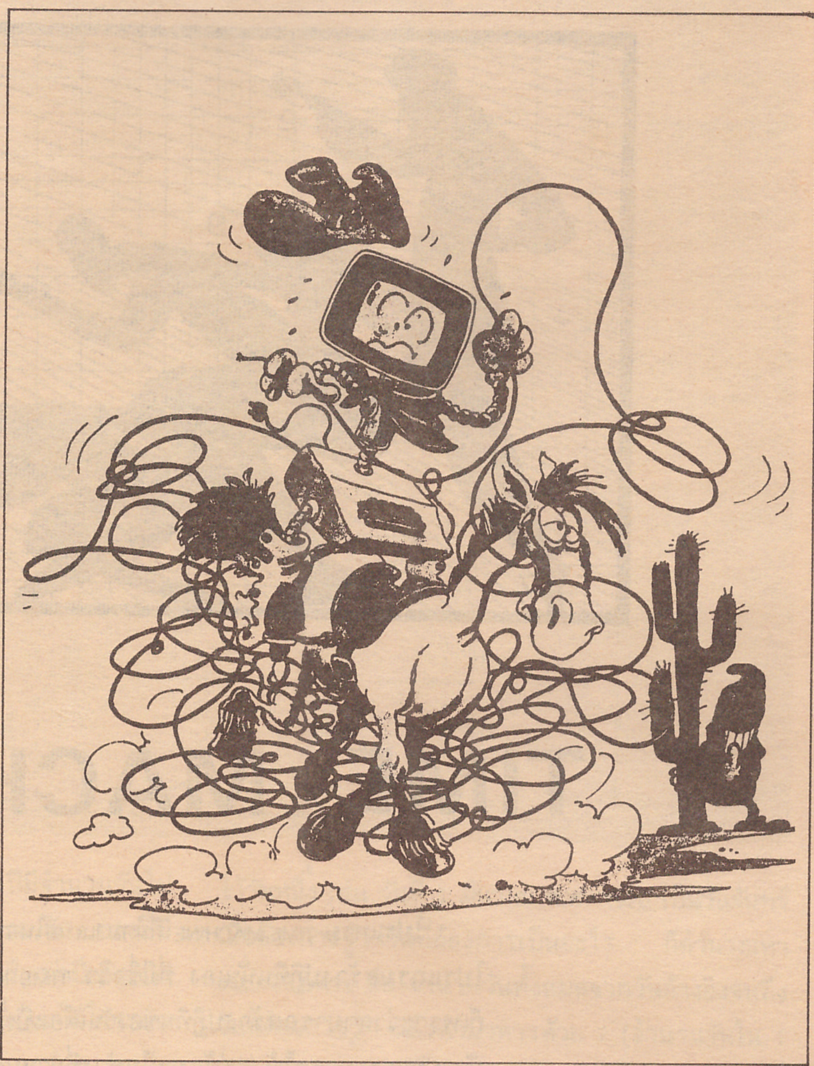
● **DEFFN** ใน Applesoft คำสั่งนี้จะมีผลขณะ execute มาถึง ดังนั้นจึงอาจมีการ DEFINE Function เดียวกันได้หลายครั้ง ในแต่ TASC จะมีการ scan หา function ที่ define ทั้งหมดในตอน compile และจัด

object code ให้ตามนั้น ดังนั้น function หนึ่งๆ จึง define ได้ครั้งเดียวและมีผลไปตลอดโปรแกรม จะเปลี่ยนแปลงหรือ define function เดิมซ้ำอีกไม่ได้

● **DIM** ใน Applesoft อาจทำการ DIMension Array ได้ 3 วิธีคือใช้ค่าคงที่ ตัวแปรหรือตามที่ default โดย interpreter แต่ใน TASC นั้น Array จะถูก demension ในขณะ compile (ไม่ว่าคำสั่ง DIM จะอยู่ที่ไหนในโปรแกรม ก่อนหรือหลังArray ที่ dimension ก็ตาม TASC จะ scan หาจนเจอและจัดการจองที่เป็น Array ให้ก่อน) ดังนั้นจึงต้อง dimension เป็นค่าคงที่แต่ถ้าไม่แน่ใจว่าจะใช้ array ขนาดเท่าไรก็ให้ dimension เผื่อไว้มากที่สุดเท่าที่จะทำได้ (เท่าที่ม memory ให้)

● **CTRL-C** การกดปุ่ม CTRL พร้อมกับ C เพื่อหยุดการทำงานของโปรแกรมนั้นจะใช้ได้เฉพาะ กับ interpreter เพราะ interpreter จะตรวจสอบว่ามีการกด CRTL-C หรือไม่ก่อน execute แต่ละคำสั่ง แต่ในโปรแกรมที่ compile โดย TASC จะไม่มีการตรวจสอบเช่นนั้น โปรแกรมจะไม่รับรู้การกด CTRL-C เลยขณะที่ execute คำสั่งต่างๆ (นอกจากคำสั่ง INPUT)

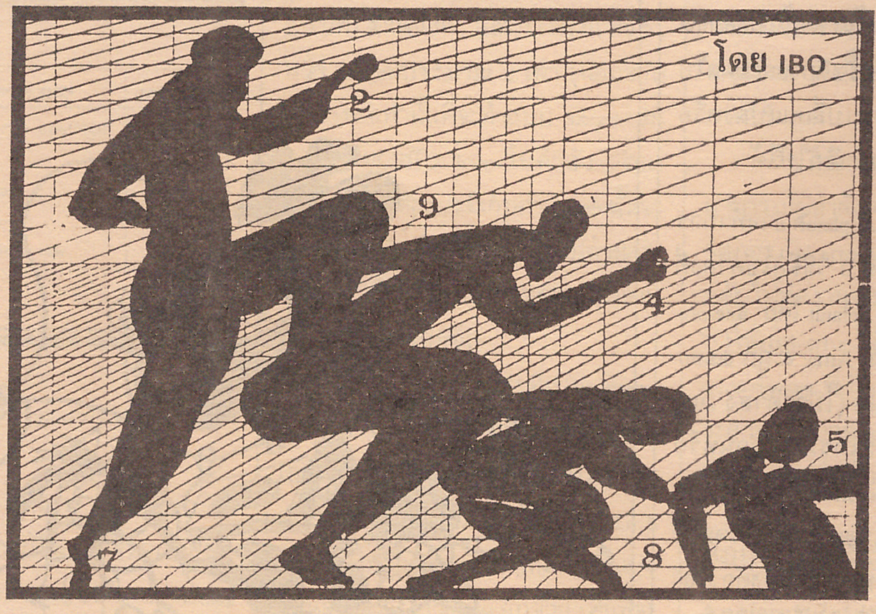
ในขณะที่มีการอ่านข้อมูลจากคีย์บอร์ดโดยคำสั่ง INPUT ถ้าเป็น interpreter เมื่อกด CTRL-C แล้ว RETURN จะเป็นการบอกให้โปรแกรมหยุดทำงาน (Halt) ซึ่งโปรแกรมที่ได้มาจากการ compile จะตรวจสอบ CTRL-C ในกรณีรับ INPUT นี้เช่นกัน โดยจะหยุดทำงานและทำการ initialize ค่าต่างๆ ใน zero page ของ memory เพื่อให้เป็นอย่างเดิมก่อน



การ BRUN (เมื่อทำการ BRUN โปรแกรมจากการ Compile จะเริ่มต้นด้วยการ initialize ค่าต่างๆ ใน zero page ให้เปลี่ยนไปตามความเหมาะสมในการทำงาน ก่อนจะเริ่มทำงานต่อไป) เพื่อให้ Applesoft interpreter กลับมาทำงานได้อย่างถูกต้อง

อีกวิธีหนึ่งที่จะหยุดการทำงานก็คือการกดปุ่ม RESET วิธีนี้จะหยุดการทำงานของโปรแกรมที่ compile ได้โดยทันทีและจะไม่มีการกลับไป initialize ค่าต่างๆ ใน zero page ให้เป็นอย่างเดิม ดังนั้นในกรณีนี้เมื่อ Applesoft interpreter กลับมาทำงานต่อก็จะเกิดความผิดพลาดขึ้นได้ เช่นอาจไปทำลาย DOS หรือทำความผิดพลาดอื่นๆ ทางแก้ในกรณีนี้คือการใช้คำสั่ง NEW เป็นคำสั่งแรกหลังจากกด RESET เพื่อให้มีการ initialize ค่าต่างๆ ใน zero page ใหม่ก่อนจะทำงานอื่นต่อไป การเกิด error ในขณะที่ RUN โปรแกรมก็จะทำให้โปรแกรมที่ได้จากการ compile หยุดทำงานโดยไม่มีการ initialize ค่าใน zero page เช่นเดียวกัน

โปรดติดตามตอนจบฉบับหน้า

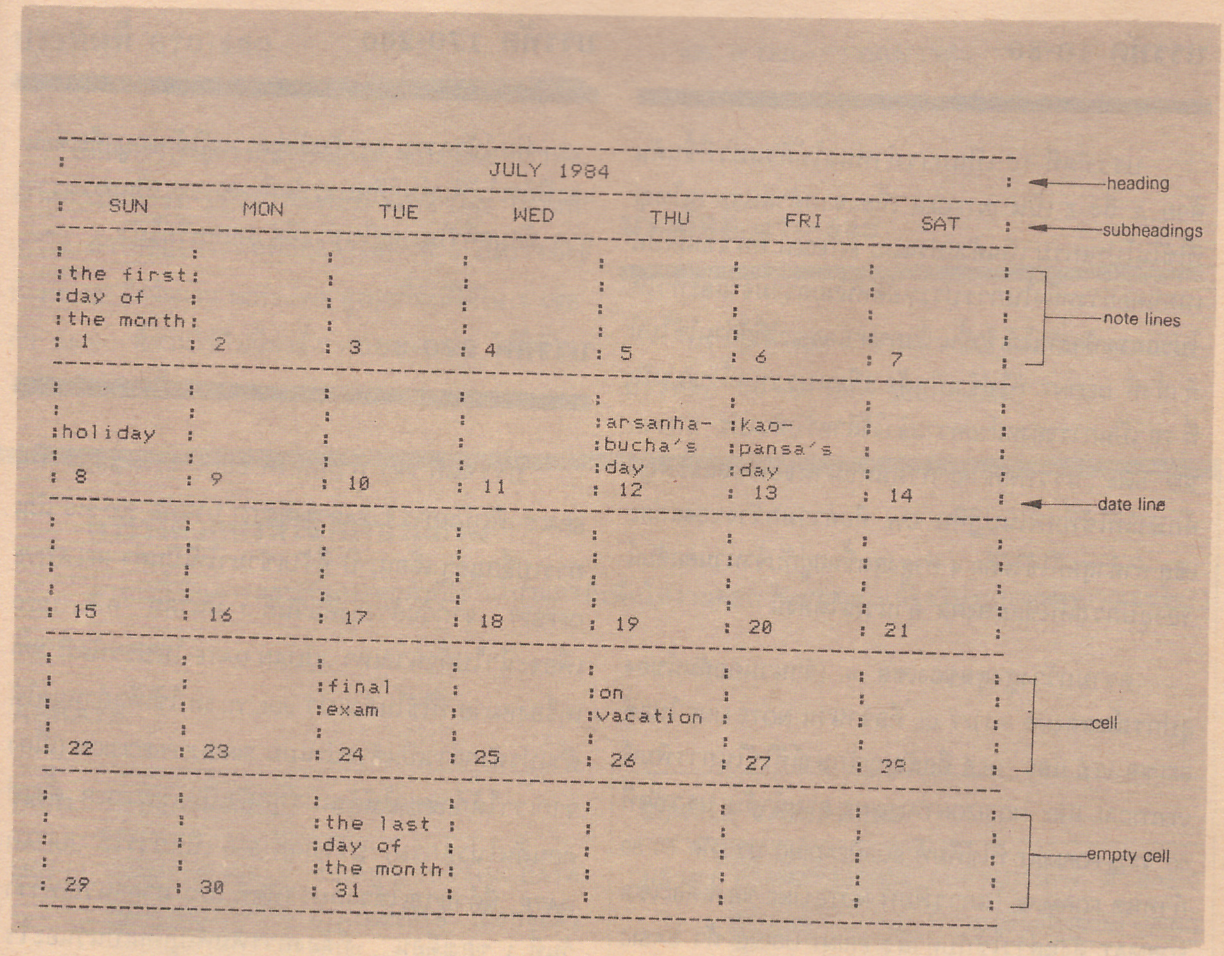


TIME- MACHINE

โปรแกรม TIME MACHINE ที่นำมาเสนอในฉบับนี้ก็คือโปรแกรมสร้างปฏิทินนั่นเอง ที่มีชื่อว่า TIME MACHINE ก็เพราะว่าสามารถสร้างปฏิทินของวันเดือนปีทั้งในอดีต ปัจจุบันและอนาคตได้ (แต่ต้องอยู่ในช่วงที่กำหนด) โดยที่ผู้ใช้เพียงแต่ป้อนข้อมูลคือ เดือนและปีที่ต้องการเข้าไป ก็จะคำนวณและสร้างปฏิทินออกมาได้ทันที ผู้ใช้ไม่จำเป็นต้องกำหนดว่า จำนวนวันในเดือนนั้นมีค่าเท่าไร วันที่เท่าไรเป็นวันอะไร ส่วนนี้โปรแกรมจะคำนวณให้เสร็จเรียบร้อย นอกจากนี้ยังสามารถเขียนโน้ตสั้น ๆ ลงในแต่ละวันได้ รูปร่างหน้าตาของปฏิทินนี้เป็นอย่างไรเชิญดูได้จากรูป

จากรูปจะเห็นว่าปฏิทินถูกแบ่งออกเป็นส่วน ๆ เป็นช่องสี่เหลี่ยม ซึ่งภายในจะมี NOTE LINE และวันที่อยู่ ต่อไปจะขอเรียกแต่ละส่วนว่า cell และเซลต่าง ๆ ที่ไม่มีวันที่ใด ๆ อยู่เรียกว่า empty cell

ในโปรแกรมนี้มีส่วนที่น่าสนใจอยู่อย่างหนึ่งก็คือการคำนวณว่าวันที่ 1 ของแต่ละเดือนเป็นวันอะไร เพื่อที่จะได้พิมพ์ปฏิทินออกมาได้ตรงกับความเป็นจริง เป็นการ



ยากที่จะคำนวณได้เนื่องจาก ในแต่ละปีมีจำนวนวันไม่เท่ากัน คือในบางปีเดือนกุมภาพันธ์มี 28 วัน บางปีมี 29 วัน เรียกว่า LEAP YEAR หรือที่ทางไทยเรียกว่าปีอธิกมาศนั่นเอง เราสามารถรู้ได้ว่าปีใดเป็น LEAP YEAR ได้โดยหลักเกณฑ์ดังนี้

1. เป็นปี (คศ.) ที่หารด้วย 4 ลงตัว แต่ต้องไม่ใช่ปีเริ่มต้นของศตวรรษหรือปีที่หารด้วย 100 ลงตัว

2. ปีที่หารด้วย 100 ลงตัว อาจเป็น LEAP YEAR ได้ก็ต่อเมื่อปีนั้นหารด้วย 400 ลงตัว

ตัวอย่างเช่น ปีคศ. 1984 หารด้วย 4 ลงตัวและหารด้วย 100 ไม่ลงตัวถือว่าเป็น LEAP YEAR ปี 1900 หารด้วย 4 ลงตัว แต่ว่าหารด้วย 100 ลงตัวด้วยจึงไม่ใช่ LEAP YEAR ปี 2000 หารด้วย 400 ลงตัวจึงถือว่าเป็น LEAP YEAR

การคำนวณว่าวันที่ 1 ของแต่ละเดือนเป็นวันอะไรนั้นอาจทำได้โดยเก็บข้อมูลของแต่ละปีเอาไว้เลย แต่วิธีนี้เป็นวิธีที่ยุ่งยากและต้องใช้เนื้อที่ในการเก็บข้อมูลมหาศาล ยังมีวิธีง่าย ๆ อีกวิธีซึ่งนิยมใช้กันคือ ต้องกำหนดวันอ้างอิงไว้วันหนึ่ง เช่น วันที่ 1 มีนาคม ค.ศ. 1920 เป็นวันจันทร์ เมื่อต้องการรู้ว่าวันที่ต้องการเป็นอะไร ก็คำนวณหา JULIAN DATE คือ จำนวนวันทั้งหมดจากวันอ้างอิงจนถึงวันที่ต้องการรู้ แล้วนำมาหารด้วย 7 (จำนวนวันใน 1 สัปดาห์) จากเศษที่ได้จากการหาจะทำให้ทราบว่าวันที่ต้องการเป็นวันอะไร โปรแกรม TIME MACHINE ก็ใช้วิธีนี้ ซึ่งช่วงที่จะคำนวณได้ตั้งแต่ 1 มีนาคม 1920 จนถึง 30 พฤศจิกายน 2009 เป็นเวลาทั้งหมดประมาณ 89 ปี

โปรแกรม

โปรแกรมนี้เขียนด้วย MICROSOFT BASIC. หรือ MBASIC แต่อย่างไรก็ตามผู้อ่านอาจนำไปใช้กับภาษา BASIC ของ VERSION อื่น ๆ ได้โดยดัดแปลงเพียงเล็กน้อย ซึ่งได้อธิบายไว้แล้ว ผู้เขียนขออธิบายหลักการทำงานของโปรแกรมอย่างคร่าว ๆ ดังนี้

บรรทัด 10-80

บรรทัดที่ 10 เป็นการกำหนดให้ตัวแปรที่ขึ้นต้นด้วย A ถึง Z เป็น INTEGER เนื่องจากตัวแปรส่วนใหญ่ที่ใช้ในโปรแกรม เป็นตัวแปรแบบ INTEGER ที่ทำเช่นนี้เป็นเทคนิคอย่างหนึ่งในการประหยัดหน่วยความจำและทำให้โปรแกรมทำงานได้เร็วขึ้น ถ้าภาษา BASIC ที่ผู้อ่านใช้ไม่มีคำสั่งที่ DEFINT ก็ไม่จำเป็นต้องเขียนบรรทัดนี้ บรรทัดที่ 20 เป็นการจองหน่วยความจำไว้สำหรับตัวแปร STRING 500 byte บรรทัดที่ 30 กำหนดขนาดของเซลและสัญ-ลักษณ์ที่ใช้พิมพ์ในปฏิทิน CW คือความกว้างของแต่ละเซล ค่าต่ำสุดที่ใช้ได้คือ 4 ค่าสูงสุดขึ้นอยู่กับจำนวนคอลัมน์ของอุปกรณ์ที่ใช้แสดงผล คำนวณได้ดังนี้

ความกว้างสูงสุดของเซล = (จำนวนคอลัมน์ของอุปกรณ์แสดงผล −1)/7 CL คือจำนวน NOTE LINE ในแต่ละเซล VL$ และ HL$ คือสัญญลักษณ์ที่ใช้ในการพิมพ์ VERTICAL และ HORIZONTAL LINE ตามลำดับ บรรทัดที่ 50 ระบุ FORMAT ที่ใช้พิมพ์ SUBHEADING บรรทัด 60-80 กำหนด FORMAT ในการพิมพ์ NOTE LINE ของแต่ละเซล FORMAT ดังกล่าวใช้กับคำสั่ง PRINT USING ถ้า BASIC ที่ท่านผู้อ่านใช้ไม่มีคำสั่ง PRINT USING ก็ไม่ต้องเขียนบรร-ทัดที่ 60-80 ฟังก์ชัน STRING$ (CW-1, " ") ในบรรทัดที่ 60 ทำให้เกิด STRING ซึ่งประกอบด้วย SPACE มีความยาวเท่ากับ CW-1

บรรทัด 90-160

C( ) เป็นตัวแปรที่แสดงว่า เซลดังกล่าวเป็น empty cell หรือไม่ เช่น จากรูปเซลที่ต่อจากเซลที่ 31 เป็น empty cell ค่า I มีได้ถึง 42 เนื่องจากขนาดใหญ่ที่สุดของปฏิทินคือ 6 × 7 cell ค่า I เรียงจากซ้ายไปขวาและบนลงล่างในปฏิทิน เซลที่เป็น empty cell C ( ) = 0

NL$ (,) เป็นตัวแปรที่เก็บข้อความที่จะพิมพ์ใน NOTE LINE NL$ (1, 2) หมายถึง เซลที่ 1 NOTE LINE ที่ 2 จากรูปจะเห็นว่าข้อความนั้นก็คือ "the first" บรรทัดที่ 100-160 เป็นการอ่าน ชื่อวัน ชื่อเดือน และจำนวนวันในแต่ละเดือน

บรรทัด 170-260

พิมพ์ข้อความ และรับข้อมูลจากผู้ใช้ ข้อมูลคือเดือนและปีที่จะให้พิมพ์ปฏิทินบรรทัดที่ 230-260 เป็นการตรวจดูว่า ข้อมูลที่รับมาอยู่ในช่วงที่กำหนดหรือไม่

บรรทัด 270-360

บรรทัดที่ 280 กำหนดให้ C(I) = 0, NL$(I,J) เป็น SPACE เพื่อเป็นการ CLEAR ข้อมูลเก่า บรรทัด 300-310 เป็นการเปลี่ยนค่า เดือน, ปี ที่รับเข้ามาให้เป็นค่า RELATIVE OFFSET จากวันอ้างอิงคือวันที่ 1 มีนาคม ค.ศ. 1920 เพื่อจะนำไปคำนวณหา JULIAN DATE ในบรรทัดที่ 320 เครื่องหมาย ที่ใช้ในตัวแปร JD!, Y!, M! ใช้เพื่อกำหนดให้ตัวแปรดังกล่าวเป็นตัวแปรแบบ SINGLE PRECISION (เนื่องจากเราได้กำหนดให้ตัวแปรทุกตัวเป็น INTEGER ตั้งแต่ตอนต้นโปรแกรม) บรรทัดที่ 330 เป็นการใช้ JULIAN DATE ที่คำนวณได้จากบรรทัดที่ 320 มาคำนวณหาว่าวันที่ 1 ของเดือน ที่ต้องการพิมพ์ปฏิทินเป็นวันอะไร ฟังก์ชัน MOD (Modulo) เป็นฟังก์ชันที่ใช้ในการหาเศษที่เกิดจากการหาร (JD! + 1) ด้วย 7 บรรทัดที่ 340 หาว่าปีที่จะพิมพ์เป็น LEAP YEAR หรือไม่ บรรทัดที่ 350 หาจำนวนวันในเดือนนั้น บรรทัดที่ 360 กำหนดวันที่ลงใน cell ที่ตรงกัน และ cell ที่ไม่ได้ใช้ให้มีค่าเท่ากับศูนย์

สำหรับฟังก์ชัน MOD ถ้า BASIC ที่ท่านผู้อ่านใช้ไม่มีคำสั่งนี้ให้เปลี่ยนแปลงโปรแกรมดังนี้ (บรรทัดที่มีในโปรแกรมให้แก้ไข บรรทัดที่ไม่มีในโปรแกรมให้เพิ่มเข้าไป)

```
330 Q1!=JD!:Q2=7:GOSUB 800:WD=Q3
340 Q1!=Y:Q2=4:GOSUB 800:Y2=Q3
342 Q1!=Y:Q2=100:GOSUB 800:Y3=Q3
344 Q1!=Y:Q2=400:GOSUB 800:Y4=Q3
346 IF (Y2=0 AND Y3<>0)OR Y4=0 THEN LP=1
    ELSE LP=0
800 Q3=INT ((Q1!/Q2-INT(Q1!/Q2)) * Q2 + 5E-6)
810 RETURN
```

บรรทัดที่ 370-460

เป็นการรับข้อความที่จะพิมพ์ใน NOTE LINE บรรทัดที่ 380 เป็นการถามผู้ใช้ว่าจะพิมพ์ NOTE LINE ในวันที่เท่าใด ถ้าไม่ต้องการพิมพ์อีกแล้วให้พิมพ์ 0 ในบรรทัดที่ 410 NB คือจำนวน NOTE LINE สูงสุดที่จะมีได้ใน 1 เซล ถ้า BASIC ที่ท่านใช้ไม่มีคำสั่ง PRINT USING ให้แก้ไขโปรแกรม ดังนี้

```
435 IF TXS= """" THEN TX$= " "
440 NL$(WD+SD,I) = LEFT$(TX$,CW-1)
```

บรรทัดที่ 470-560

บรรทัดที่ 480 คำนวณค่า TAB และพิมพ์ HEADING
บรรทัดที่ 520-560 พิมพ์ SUBHEADING

ตารางตัวแปรที่ใช้ในโปรแกรม

ตารางนี้เป็นการสรุปตัวแปรทุกตัวที่ใช้ในโปรแกรม เพื่อให้ง่ายต่อการแกะและแก้ไขโปรแกรม หมายเลขใน ( ) คือ line number

ARRAYS

LABEL	CONTENTS	COMMENTS
MO$ (,)	NAME OF THE MONTHS	ชื่อเดือนเก็บไว้ใน DATA STATEMENTS เลือกใช้ตัวพิมพ์ใหญ่ (UPPERCASE) หรือตัวพิมพ์เล็ก (LOWERCASE) ได้ตามต้องการ
MD ( )	LENGTH OF EACH MONTH	สำหรับเดือนกุมภาพันธ์ = 28 และจะบวกด้วย 1 สำหรับ LEAPYEAR (340)
C ( )	IMAGE OF CALENDAR	C(I) = O สำหรับ EMPTY CELL เซลที่ไม่ใช่ EMPTY CELL C(I) มีค่าตั้งแต่ 1-31
NL$(,)	NOTE LINES FOR EACH CELL	ค่า SUBSCRIPT อันแรกบอกหมายเลขของเซล ค่า SUBSCRIPT อันที่สองบอกว่าเป็น NOTE LINE ที่เท่าไร
WY$ ( )	NAMES OF THE WEEKDAYS	ชื่อวันเก็บใน DATA STATEMENT เลือกใช้ตัวพิมพ์ใหญ่หรือตัวพิมพ์เล็กตามต้องการ

SIMPLE VARIABLES

CN$,MF$	FORMAT STRING TO USE	ลบตัวแปรนี้ทิ้ง ถ้า BASIC ที่ท่านใช้ไม่มี PRINT USING (620,680) แก้ไขโปรแกรมดังที่อธิบายไว้แล้ว
I,J	COUNTER VARIABLES	
M,Y	MONTH AND YEAR INPUT FORM KEYBOARD	M มีช่วงตั้งแต่ 1-12 Y ตั้งแต่ 1920-2009 (220)
D	BASE DATE, ALWAYS EQUAL TO 1 (THE FIRST DAY OF THE MONTH)	
Y!	RELATIVE OFFSET OF YEAR Y FROM BASE YEAR (1920)	
M!	RELATIVE OFFSET OF MONTH Y FORM BASE MONTH (MARCH)	! แสดงว่าเป็น SINGLE PRECISION

CW	CELL WIDTH	ค่าต่ำสุด = 4, CW × 7 + 1 คือความกว้างของปฏิทิน เลือกค่าให้พอดีกับอุปกรณ์แสดงผลที่ใช้
CL	CELL LENGTH	ค่าต่ำสุด = 2 เลือกค่าให้พอดีกับอุปกรณ์แสดงผล
VL$,HL$	CHARACTERS USED FOR VERTICAL AND HORIZONTAL LINES.	อาจเปลี่ยนแปลงได้ตามความเหมาะสม
PW	OVERALL CALENDAR WIDTH	
RS	HORIZONTAL RULE	
NB	NUMBER OF NOTE LINES PER CELL	
OF	OFFSET TO PLACE DAY-OF-WEEK SUB HEADINGS OVER THEIR COLUMNS.	
BC$	TEXT TO PRINT FOR BLANK CELL DATE LINE	
WD	DAY OF WEEK OF THE FIRST OF THE MONTH	มีตั้งแต่ 0-6 (0 วันอาทิตย์ 1 วันจันทร์ 2......) (330)
LP	LEAP YEAR FLAG	0 = non leap year, 1 = leap year
LW	LAST WEEK NUMBER	
SD	DATE TO BE ANNOTATED INPUT FROM KEYBOARD	
LD	LAST DAY IN MONTH M	
TXS	TEXT OF NOTE LINE, INPUT FROM KEYBOARD	
HS	MONTH-YEAR CALENDAR HEADING	
LH	LENGTH OF MONTH-YEAR HEADING	
TB	COMPUTE TAB POSITION	ใช้พิมพ์ heading (480-490) พิมพ์ SUBHEADING (540-540) พิมพ์ NOTE LINES (670-620) พิมพ์ DATE LINES (660-680)

บรรทัดที่ 570-720

เป็นการพิมพ์ตัวปฏิทินส่วนที่เหลือทั้งหมด ถ้า BASIC ที่ท่านใช้ไม่มีคำสั่ง PRINT USING ให้แก้ไขโปรแกรมดังนี้

```
620 PRINT TAB(TB) VL$;NL$(DA+(W-1) * 7,ML);
680 PRINT TAB(TB) LV$;C(DN);
700 PRINT TAB(TW-1) VL$:PRINT R$
```

อธิบายโปรแกรมมาจนจบแล้ว หวังว่า ผู้อ่านคงพอเข้าใจการทำงานของโปรแกรมกันบ้างแล้ว ถ้าท่านจะนำโปรแกรมนี้ไปใช้ อย่าลืมเปลี่ยนแปลงค่า cell width CW และ cell length CL ให้เหมาะสม กับอุปกรณ์ที่ใช้แสดงผล และถ้าจะพิมพ์ออกเครื่องพิมพ์ละก็ ให้แก้คำสั่ง PRINT ตั้งแต่บรรทัดที่ 470 ถึง 720 ให้เป็น LPRINT ให้หมด ท่านก็จะได้ปฏิทินทำเองขนาดใหญ่เล็ก แล้วแต่จะกำหนดตามต้องการ สำหรับท่านที่ใช้เครื่อง APPLE ถ้าต้องการให้พิมพ์ออกเครื่องพิมพ์ ให้เติมคำสั่ง PR=m ในบรรทัดที่ 465 และ PR=n ในบรรทัดที่ 715 (m คือ SLOT NUMBER ที่ใส่ print INTERFACE CARD n คือ SLOT NUMBER ที่ใส่ VIDEO INTERFACE CARD)

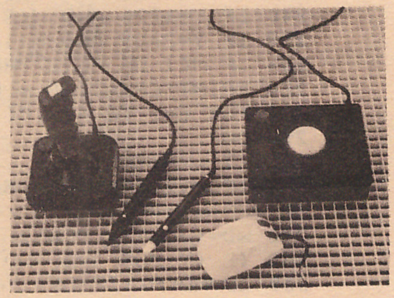
ในฉบับต่อ ๆ ไป ผู้เขียนจะพยายามหาโปรแกรมที่มีแนวความคิดแปลก ๆ และมีเทคนิคที่น่าสนใจมาเสนอกันอีก หวังว่าคงเป็นประโยชน์ทั้งด้านสาระและบันเทิงแก่ผู้อ่านบ้าง สำหรับฉบับนี้ก็ต้องลากันแค่นี้ก่อนนะครับ ■

```
10 DEFINT A-Z
20 CLEAR 500
30 CW=10:CL=5:VL$=":":HL$="-"
40 PW=CW*7+1:R$=STRING$(PW,HL$):NB=CL-1
50 OF=INT((CW-3)/2)+1
60 BC$=VL$+STRING$(CW-1," ")
70 CN$=VL$+STRING$(CW-3," ")+"##"
80 MF$=VL$+"\"+STRING$(CW-3," ")+"\"
90 DIM MO$(12),MD(12),C(42),NL$(42,NB),WY$(7)
100 FOR I=1 TO 12:READ MO$(I):NEXT I
110 DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
120 DATA AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
130 FOR I=1 TO 12:READ MD(I):NEXT I
140 DATA 31,28,31,30,31,30,31,31,30,31,30,31
150 FOR I=1 TO 7:READ WY$(I):NEXT I
160 DATA SUN,MON,TUE,WED,THU,FRI,SAT
170 PRINT"THE TIME MACHINE: A CALENDER GENERATOR"
180 PRINT"--- ---- -------  - -------- ---------"
190 PRINT"INPUT THE CALENDER MONTH AS:MONTH,YEAR,"
200 PRINT"FOR EXAMPLE:'1,1984' FOR JANUARY,1984"
210 PRINT"VALID MONTHS ARE 3,1920 THROUGH 11,2009"
220 INPUT"NOW TYPE THE MONTH,YEAR";M,Y
230 IF Y<1920 OR Y>2009 THEN 210
240 IF Y=1920 AND M<3 THEN 210
250 IF Y=2009 AND M>11 THEN 210
260 IF M<1 OR M>12 THEN 210
270 PRINT"CREATING A CALENDER FOR ";MO$(I);Y
280 FOR I=1 TO 42 :C(I)=0:FOR J=1 TO NB:NL$(I,J)="":NEXT J:NEXT I
290 D=1
300 Y1!=Y-1920
310 IF M>2 THEN M1!=M-3 ELSE M1!=M+9:Y1!=Y1!-1
320 JD!=INT(1461*Y1!/4)+INT((153*M1!+2)/5)+D-1
330 WD=(JD!+1) MOD 7
340 IF (Y MOD 4=0 AND Y MOD 100<>0) OR Y MOD 400=0 THEN LP=1 ELSE LP=0
350 LD=MD(M):IF LP=1 AND M=2 THEN LD=29
360 FOR I=1 TO LD:C(I+WD)=I:NEXT I
370 PRINT"TO ANNOTATE A DATE,TYPE IN THE DATE (1-"LD");"
380 INPUT "TYPE 0 WHEN FINISHED";SD
390 IF SD<1 OR SD>LD THEN 470
400 PRINT"ENTER UP TO"NB"LINES. MAXIMUM LINE LENGTH IS"CW-1
410 FOR I=1 TO NB
420 PRINT"ENTER LINE";I;
430 TX$="":INPUT TX$
440 NL$(WD+SD,I)=TX$
450 NEXT I
460 PRINT"TEXT STORED.":GOTO 370
470 PRINT:PRINT:PRINT R$
480 H$=MO$(M)+STR$(Y):LH=LEN(H$):TB=INT((PW-LH)/2)+1
490 PRINT VL$ TAB(TB) H$ TAB(PW-1) VL$
500 PRINT R$
510 PRINT VL$;
520 FOR DA=1 TO 7
530 TB=(DA-1)*CW+OF
540 PRINT TAB(TB) WY$(DA);
550 NEXT DA
560 PRINT TAB(PW-1) VL$:PRINT R$
570 LW=-INT(-(LD+WD)/7)
580 FOR W=1 TO LW
590 FOR ML=1 TO NB
600 FOR DA=1 TO 7
610 TB=(DA-1)*CW
620 PRINT TAB (TB) USING MF$;NL$(DA+(W-1)*7,ML);
630 NEXT DA:PRINT TAB(PW-1)VL$:NEXT ML
640 FOR DA=1 TO 7
650 TB=(DA-1)*CW
660 DN=DA+(W-1)*7
670 IF C(DN)=0 THEN PRINT TAB(TB) BC$;:GOTO 690
680 PRINT TAB(TB) USING CN$;C(DN);
690 NEXT DA
700 PRINT VL$:PRINT R$
710 NEXT W
720 END
```

มีอะไรในไมโครฯ ฉบับหน้า (สิงหาคม 2527)



- HP-150 คอมพิวเตอร์ touch screen ที่คุณชี้นิ้วสั่งได้
- MS-DOS 2.0 ตอนที่ 2
- Special Report : Network
  ต่อไมโครกับไมโครด้วย LAN
  เมื่อไมโครต่อกับเมนเฟรม
- iAPX 188 (80188) 8088 ไมโครโปรเซสเซอร์รุ่นใหม่ของ intel



- ภาษาใหม่ของไมโครฯ แนะนำให้คุณรู้จัก FORTH, MODULA-2 และภาษาอื่น ๆ ที่คุณอาจไม่เคยรู้จักมาก่อน
- Pointer : joystick, mouse, light pen, Graphic tablet ล้วนเป็นตัวชี้ (pointer) ทั้งสิ้น
- Software Review : Word Processor ภาษาไทย
- คอลัมน์ประจำครบครัน

ใบสมัครสมาชิก/ต่ออายุสมาชิก

ชื่อ..นามสกุล..

ที่อยู่..รหัสไปรษณีย์..........................

มีความประสงค์ ☐ สมัครสมาชิก ☐ ต่ออายุสมาชิกหมายเลข.........................

วารสาร **เปิดโลก**

สมาชิก	กระดาษธรรมดา	☐ 6 เดือน 80 บาท	☐ 1 ปี 150 บาท	☐ 2 ปี 280 บาท
	กระดาษปอนด์	☐ 6 เดือน 90 บาท	☐ 1 ปี 180 บาท	☐ 2 ปี 360 บาท

วารสารไมโครอีเลคทรอนิคส์

สมาชิก	กระดาษธรรมดา	☐ 6 เดือน 100 บาท	☐ 1 ปี 200 บาท	☐ 2 ปี 380 บาท
	กระดาษปอนด์	☐ 6 เดือน 120 บาท	☐ 1 ปี 240 บาท	☐ 2 ปี 460 บาท

เริ่มรับตั้งแต่ฉบับที่.....................

ส่งธนาณัติ หรือเช็คไปรษณีย์สั่งจ่าย ปท. **ราชเทวี** หรือเช็คธนาคารขีดคร่อมเข้าบัญชีในนามของ "บริษัท ไมโครมีเดีย จำกัด" 248 สยามสแควร์ ซอย 2 หลังโรงภาพยนตร์ลิโด ถ.พระราม 1 กรุงเทพฯ 10500 โทร. 2514229